MSXテクニカルガイドブック
ディスク編

目次

序文

　ディスクドライブ内蔵機が普及した現在、これを読んでいる皆さんの大部分はディスク装置を利用していることと思います。今日ではディスクの使用できない環境が考えられなくなっているのと同時に、ソフト開発においてディスクアクセスを避けて通ることはできません。

　この本は、プログラム開発の際に必要なディスクアクセスに関する情報を利用しやすくまとめた物です。プログラム開発の際に御活用ください。また、未公開になっているディスクドライバ関係の解析資料も併せて掲載しました。こちらは一般的には必要な情報ではありませんが、ディスクアクセスに関してのさらなる理解の助けとなる事を願います。

お断わり

　この本は、最低限MSX2テクニカルハンドブックを読者が持っていることを前提に書いてあり、基本的な内容については省略してある場合があります。それらについては、MSX2テクニカルハンドブック等を参照して下さい。また、ページ数等の都合により、割愛した情報もありますので、本書によってわからない部分は各文献を参照して下さい。

　この本の特に第5部では、未公開になっている部分、異機種間で互換性のない部分が含まれています。そして、これら未公開情報を一般的に配布、販売するソフトで使用することは、絶対に避けるようにしてください。もしこれらの情報を使用した場合、MSXの最大の特徴である互換性を欠くことになってしまいます。十分御注意下さい。

　独自の解析資料は制作に当たって十分注意していますが、間違いの可能性がないとはいえませんのでご了承下さい。

　本書の内容を利用したことによるいかなる事態にも当方は一切の責任を負いません。

　なお、この本の制作に当たって株式会社アスキーはまったく関与していません。この本の内容に関する疑問、質問等はすべて筆者までおねがいします。

凡例

本書では以下のような表記を使用する場合があります。

表記	意味
DOS1	MSX-DOS Ver.1.03 以前
DOS2	MSX-DOS Ver.2.10 以降
MSX1	MSX BASIC Ver.1.0 を搭載した MSX
turboR	MSX TurboR
Datapack1/2/3	MSX-Datapack Volume 1/2/3
(XXXXXX,n)	XXXXXX の示すアドレスからｎバイトの内容
(YYYYYY,bn)	YYYYYY の示すアドレスの内容のビットｎ
/a/b/c/d/e/f/g/h/	１バイトをビット単位で内容を表示したもの a:ビット７の内容・・・h:ビット０の内容

参考文献

ＭＳＸ２テクニカルハンドブック(アスキー)

ＭＳＸｄａｔａｐａｃｋ(アスキー)

ＭＳＸｄａｔａｐａｃｋｔｕｒｂｏＲ版(アスキー)

日本語ＭＳＸ－ＤＯＳ２リファレンスマニュアル(アスキー)

ＨＤ　Ｉｎｔｅｒｆａｃｅ　ユーザーズマニュアル(アスキー)

ＭＳＸテクニカルデータブック増補改訂版(アスキー)

ＭＳＸテクニカルガイドブック(アスキャット)

フロッピ・ディスク装置のすべて(ＣＱ出版社)

プロテクト活用ハンドブック(誠文堂新光社)

バックアップ活用テクニック(三才ブックス)

第１部 ＭＳＸとディスクシステム

第１章　MSX のディスクシステム

MSX には、そのその高い拡張性の一部としてディスクシステムが存在します。もちろんその仕様は論理フォーマットレベル、アクセスレベルで統一され、どのメーカーのディスクドライブでも共通に利用できるようになっています。これによって、たとえメディアの種類が異なっていても同じ方法でアクセスできることを保証しています。ただし、ハードウェアについては規格として定まっていないため、ソフトウェアが直接 I/O ポートや FDC を通してディスクをアクセスすることは禁止されています。

ディスクとの入出力は、ディスクシステムの中核を成す DOS カーネルが担当します。ドライブのハードウェアによる差異は、各ディスクインターフェイスに付随するディスクドライバが吸収するようになっています。

標準の DOS である MSX-DOS は、16 ビットパソコンで主流になっている MS-DOS に準じたフォーマットを採用し、その上、8 ビットパソコンで発展した CP/M80 のソフトがメディアコンバートさえすればほとんどそのまま動くようになっています。

また、Disk BASIC は DOS の機能を利用しているため、DOS と Disk BASIC でフォーマットが同一になっています。

また、その仕様を受け継いで発表された MSX-DOS2 は、サブディレクトリへの対応など MSX-DOS の機能を大幅に拡張しています。

ディスクドライブは MSX1、MSX2 ではオプション装置となっていました。しかし、MSX2+以降では仕様としてディスクドライブ装置が加えられ、必ずディスクを持っていることになっています。

第２章　ディスクシステムの位置づけ

MSX におけるディスクシステムはあくまでも拡張機器の一つであるため、ディスクシステム用に特別の拡張手段が準備されている訳ではありません。しかし、ディスクドライブはシステムに密着した機器であるため、MSX で定められていないワークエリアを使用したりしています。ディスクインターフェイスは、同時に 4 つまで接続できるようになっています。それ以上のインターフェイスを接続した場合にはシステムが起動できません。

第３章　メモリマップ

MSX でディスクを使用している状況のメモリマップは、Disk BASIC 上であるか、MSX-DOS 上であるかによって大きく異なります。DOS2 では、ディスクのワークエリアが減少するため、一般にユーザーの利用できるエリアが増加します。

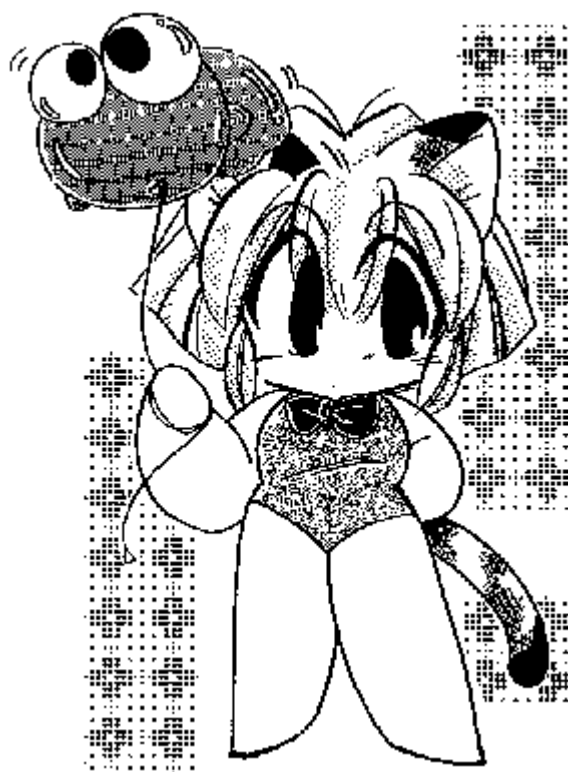
BASIC 環境の場合、MSX では DE3FH までのエリアを使用してプログラムを作成することが推奨されています。これは、2DD ドライブを接続するインターフェース１台を接続した場合のフリーエリアを想定しています。

Disk BASIC の場合

```
0000H
        BIOS
4000H
        BASIC
(BOTTOM)  RAM が 32KB 以上実装されている場合 8000H
        フリーエリア
(HIMEM)
        MSX-DOS ワークエリア
F380H
        システムワークエリア
FFFFH
```

MSX-DOS(2)の場合

```
0000H
        システムスクラッチエリア
0100H
        TPA
(0006H)
        MSXDOS.SYS
(HIMEM)
        MSX-DOS ワーク
F380H
        システムワーク
FFFFH
```

第２部Ｄｉｓｋ　ＢＡＳＩＣ

第１章　Disk BASIC Ver.1.

　ここでは、Disk BASIC のうち、特にいくつかの命令を解説します。

DSKI$(＜ドライブ番号＞,＜論理セクタ番号＞)　　　(関数)

　F351H,F352H 番地の内容が示すアドレスへドライブ番号で示すドライブの論理セクタ番号で示されるセクタのデータを読み込みます。
　ドライブ番号は 0:カレントドライブ、1:A・・・8:H です。
　論理セクタ番号は 0 から始まります。セクタ番号のチェックはしていないので注意して下さい。
　F351H,F352H の値は Disk BASIC の命令で破壊されます。

DSKO$ ＜ドライブ番号＞,＜論理セクタ番号＞　　　(ステートメント)

　F351H,F352H 番地の内容が示すアドレスからドライブ番号で示すドライブの論理セクタ番号で示されるセクタのデータを書き込みます。
　ドライブ番号は 0:カレントドライブ、1:A・・・8:H です。
　論理セクタ番号は 0 から始まります。セクタ番号のチェックはしていないので注意して下さい。
　F351H,F352H の値は Disk BASIC の命令で破壊されます。

第２章　Disk BASIC Ver.2.

　特にいくつかの命令を説明します。

CALL DOS2MEMCHK　(日本語 MSX-DOS2 カートリッジのみ)

　DOS2 の ROM とカートリッジの内蔵 RAM のチェックを行います。
　このコマンドを使用する時、確認メッセージで確認の後メモリチェックを行い、チェックの後システムは停止します。復帰はできません。

第３章　ファンクションコールの利用法

　ファンクションコールは Disk BASIC の環境からも利用することができます。Disk BASIC からコールしたい時は、DOS の場合と同じように C レジスタにファンクション番号をセットしてから F37DH 番地をコールします。
　Disk BASIC からのコールではいくつかのファンクションで制限があります。また、Disk BASIC と DOS で動作の異なるファンクションもありますが、詳しくは第３部第５章を参照して下さい。
　Disk BASIC から利用する場合は、ページ１からのコールとページ１への読み込み、書き込みができません。また、FCB をページ１に置くこともできません。

第４章　Disk BASIC エラーコード表

ディスク関係の命令で返されるエラーコードを説明します。

50 FIELD overflow	FIELD で定義したフィールドサイズの合計が 256 バイトまたは指定の値を越えた。
52 Bad file number	オープンしていないファイル番号を指定した。指定したファイル番号が MAXFILES の値を越えている。
53 File not found	指定されたファイルが見つからない。
54 File already open	OPEN で指定したファイル番号は他のファイルで使用中です。
55 Input past end	ファイル中のデータを全て読み込んだ後さらに入力しようとした。
56 Bad file name	ファイル名が適当でない
57 Direct statemen	アスキー形式のプログラムのロード中プログラム以外のデータがあった。
58 Sequential I/O only	シーケンシャルファイルに対してランダム入出力を行おうとした。
59 File not OPEN	オープンされていないファイルに対して入出力を行おうとした。
60 Bad allocation table	ディスクがイニシャライズされていない。
61 Bad file mode	シーケンシャルファイルに対して PUT、GET、LOF が使用されたり、ランダムファイルに対して LOAD が使用された。
62 Bad drive name	ドライブ名の指定が間違っている。
64 File still open	ファイルの CLOSE を忘れている。
65 File already exists	NAME で新しいファイル名として指定された名前が既に存在していた。
66 Disk full	ディスク上に空き領域がない。
67 Too many files	ディレクトリエントリがもうないのに新たにファイルを作成しようとした。
68 Disk write protected	書き込み禁止のディスクに書き込みを行った。
69 Disk I/O error	ディスク入出力中にエラーが発生した。
70 Disk offline ディスクがセットされていない。	70 Disk offline　　　　ディスクがセットされていない。
71 Rename across disk	異なるドライブ間で NAME を実行した。

第３部 ＭＳＸ－ＤＯＳ

第１章　MSX-DOS

１．１　MSX-DOS の構成

　MSX-DOS は、ROM によって供給される MSX-DOS カーネルと、ディスクによって供給される MSXDOS.SYS と COMMAND.COM の 2 つのシステムファイルによって構成されています。
　DOS カーネルは、増設ドライブのインターフェイス部分や、ドライブ内蔵機種では本体内に 16KB の ROM が搭載されています。
　DOS カーネルは生産時期によっていくつかのバージョンがあるようですが、通常このことを意識する必要はありません。またシステムファイルもいくつかのバージョンが存在します。

MSX-DOS の構成

```
DOSカーネル(ROMによって供給)
  いくつかのバージョンがあります。
MSXDOS.SYS
  ver.0.26    MSXDOS  .SYS       3072  84-01-01
  ver.1.03    MSXDOS  .SYS       2432  85-08-23  21:29:34
COMMAND.COM
  ver.0.12    COMMAND .COM       6144  84-01-01
  ver.1.11    COMMAND .COM       6656  85-09-02  22:10:16
```

１．２　MSX-DOS の起動

MSX-DOS の起動手順を説明します。

１．DISK ROM の INIT ルーチンは SHIFT キーが押されていれば BEEP 音を出してから BASIC を起動します。
２．CTRL キーが押されているか調べて BEEP 音を出します。そしてディスクシステムのためのワークエリアを確保します。さらに、各ディスク ROM のディスクドライバが設定をします。
３．H.STKE(FEDAH)に何らかのルーチンが設定されていれば、DISK-BASIC の環境を設定してから H.STKE にジャンプします。
４．TEXT エントリをもつカートリッジがあった場合、DISK-BASIC の環境を設定してからそのカートリッジの BASIC プログラムを実行します。
５．カレントドライブのブートセクタの先頭の 100H バイトの内容を C000H からに転送します。読み込めなかった場合は DISK-BASIC が起動します。
６．Cy フラグをリセットした状態で C01EH をコールします。ここにプログラムを書いておくとそのプログラムが起動します。
７．RAM 容量をチェックして、64KB の容量がなかった場合は DISK-BASIC が起動します。
８．MSX-DOS の環境が設定され Cy フラグをセットして C01EH をコールします。通常ここでは MSXDOS.SYS が 100H からにロードされ実行されます。MSXDOS.SYS が存在しない場合 DISK-BASIC が起動します。
MSXDOS.SYS は自分自身を高位アドレスに転送してから、
COMMAND.COM を 100H からにロードし実行します。
COMMAND.COM が存在しない場合は"INSERT ADISKETTE"のメッセージが表示され、正しいディスクがセットされるのを待ちます。

１．３　ファンクションコール

ファンクションコールは MSX-DOS が提供するサポートルーチン群です。MSX-DOS の基本的な入出力を行う BDOS を汎用のサブルーチンとしてまとめてあります。ファンクションは 42 個あります。ファンクションコールの詳細については第 5 章をご覧下さい。

ファンクションコールを利用する時は、C レジスタにファンクションコールのファンクション番号をセットしてファンクションコールのエントリ(0005H)をコールします。コールされると MSX-DOS は内部スタックに切り換えますので、必要なユーザーのスタックは 8 バイトです。

ファンクションコールでは全てのレジスタ(AF,BC,DE,HL)は破壊されます。DOS1 では裏レジスタ及びインデックスレジスタも破壊されます。DOS2 では裏レジスタとインデックスレジスタはそれが値を返す場合を除いて保存されます。また、CP/M との互換性のために CP/M のファンクションに対応するファンクションでは、A=L,B=H でリターンします。

第２章　MSX-DOS2

２．１　MSX-DOS2の構成

MSX-DOS2は、ROMによって供給されるMSX-DOS2カーネルと、ディスクによって供給されるMSXDOS2.SYSとCOMMAND2.COMの２つのシステムファイルによって構成されています。

DOS2カーネルは、MSX-DOS2カートリッジとして供給されるほか、MSX-DOS2が標準装備になったturboR以降の機種は本体内にROMを持っています。DOS2カーネルのROMは48KBの容量があります。48KBのROMはページ１にあり16KB*3のローカルバンクを構成しています。　DOS2カーネルはいくつかのバージョンがあります。

MSX-DOS2の構成

```
DOS2カーネル(ROMによって供給)
  ver.2.20    日本語MSX-DOS2
  ver.2.30    MSXturboR FS-A1ST
  ver.2.31    MSXturboR FS-A1GT
MSXDOS2.SYS
  ver.2.20    MSXDOS2 .SYS       4480  88-10-14  15:12:50
  ver.2.30    MSXDOS2 .SYS       4870  90-09-03  16:58:40
COMMAND2.COM
  ver.2.20    COMMAND2.COM      14976  88-10-25  15:04:24
  ver.2.30    COMMAND2.COM      15472  90-09-11  16:24:42
  ver.2.31    COMMAND2.COM      15708  91-06-19  20:11:26
```

２．２　ファンクションコール

ファンクションコールはMSX-DOS2が提供するサポートルーチン群です。

ファンクションコールはMSX-DOSでの42個に新たに50個が加わって92個あります。MSX-DOS2ではMSX-DOSであった42個のファンクションコールはほとんど互換性があると考えてよいのですが、細かい部分では変更されている部分があり、注意が必要な場合があります。ファンクションコールの詳細については第5章をご覧下さい。

ファンクションコールを利用する時は、Cレジスタにファンクション番号をセットしてファンクションコールのエントリ(0005H)をコールします。

第３章　プログラムの起動と終了

３．１　外部コマンドの起動

DOS 上で外部コマンド(COM ファイル)を実行する際の手順は次のようになっています。

コマンド行のパラメータについて、その長さをシステムスクラッチエリアの 80H 番地に実際の文字列を 0081H 以降に格納し、さらに最初の 2 つのパラメータをファイル名とみなしてそれを FCB の形式でそれぞれシステムスクラッチエリアの 005CH と 006CH 以降にそれぞれ格納します。その後で外部コマンドを 0100H 以降に読み込み、0100H にジャンプします。

005CH および 006CH からの内容はそのまま FCB としてファイルのアクセスに使用することができるようになっていますが、両 FCB が 16 バイトしか離れていないために完全な FCB として利用できるのはどちらか片方だけになります。

0080H からには、全パラメータ(外部コマンド名その物は含まれません)が入っているので、ファイル名以外のパラメータや 3 つ以上のパラメータを使用する時に利用できます。

このようにシステムスクラッチエリアの 005CH～00FFH は、DOS からパラメータを渡すために使用されますが、外部コマンドが起動した後はワークエリアとして使うこともできます。

システムスクラッチエリアの 0000H～005BH の内容を次に示します。

```
0000H,3   リブートエントリ
0003H,2   予約
0005H,3   MSX-DOS のエントリ
0008H,4   RST08H  ユーザ用
000CH,4   RDSLT ルーチンへのエントリ
0010H,4   RST10H  ユーザ用
0014H,4   WRSLT ルーチンへのエントリ
0018H,4   RST18H  ユーザ用
001CH,4   CALSLT ルーチンへのエントリ
0020H,4   RST20H  ユーザ用
0024H,4   ENASLT ルーチンへのエントリ
0028H,4   RST28H  ユーザ用
002CH,4   予約
0030H,4   CALLF ルーチンへのエントリ
0034H,4   予約
0038H,3   割り込みベクタ
003BH,33  拡張スロット切り換えコードによって使用
```

３．２　外部コマンドの終了

プログラムの実行が終了した時には、次のいずれかの方法でMSX-DOSのコマンドレベルに戻ることができます。

1. スタックポインタを変更していなければ、RETを実行する。
2. 「プログラムの終了」ファンクション(00H)を実行する。
3. 0000Hのウォームスタートのためのエントリにジャンプする。
4. 「エラーコードを伴った終了」ファンクション(62H)を実行する。(DOS2)

ただし、DOS2では外部コマンドの終了には「エラーコードを伴った終了」ファンクションを使用することになっています。しかしDOS2で上の3つの方法を使用したとしても正常に終了するので、DOS1のプログラムをDOS2で使用できないということではありません。

DOSに処理が返ると、ますMSXDOS.SYSがTPAの内側にあるCOMMAND.COMが破壊されているかどうかを調べます。破壊されていれば、再びディスクからロードします。破壊されていなければディスクからのロードは行わないのでコマンドレベルに戻るのがいくらか早くなります。

第４章　ディスクのデータ構造

ここでは、ディスクに記録されるデータがどのように記録されているかを説明し、またディスクアクセスに関係して必要となるデータ構造について解説します。

４．１　ディスクの内容

ディスクはディスク内のファイルの管理情報を記録する管理領域とファイルのデータを保存するデータ領域があります。

管理領域には、ブートセクタ、FAT、ディレクトリがあります。ブートセクタはセクタ０にあり、続くセクタにFAT、ディレクトリの順に存在します。FATとディレクトリのサイズはブートセクタの情報によって知ることができます。

各領域とセクタ番号の対応

	2DD 9セクタ	2DD 8セクタ	1DD 9セクタ	1DD 8セクタ
ブートセクタ	0	0	0	0
FAT1	1～3	1～2	1～2	1
FAT2	4～6	3～4	3～4	2
ディレクトリ	7～13	5～11	5～11	3～9
データ領域	14～1439	12～1279	12～719	10～639
クラスタ数	713	634	354	315

４．２　セクタ

　実際にデータが記録される最小の単位はセクタです。2DD の場合、セクタは 512 バイトの容量があります。ハードディスク、RAM ディスクといったフロッピーディスク以外のメディアでは物理的なフォーマットはもちろん異なりますが、論理セクタのレベルではフロッピーディスクも他のメディアも同一視できるようになっています。(セクタサイズが異なる可能性はあります。)
　1 枚の 2DD フロッピディスクには 1440 のセクタあります。ある特定のセクタは論理セクタ番号で表されます。論理セクタ番号は 0～1439 となります。セクタ番号と言った場合論理セクタ番号を指すのが普通です。
　MSX ではディスクアクセスに論理セクタを用います。これによってたとえメディアが異なっても同様のアクセスをすることが可能となっています。

４．３　クラスタ

　データ領域の管理にはクラスタが用いられます。クラスタは 1～数セクタ(2DD のディスクの場合 2 セクタ)を単位にデータ領域を管理します。クラスタは 2 から始まるクラスタ番号を持っています。クラスタ 2 がデータ領域の先頭に位置し、そこからデータ領域の最後までクラスタが存在します。
　クラスタ番号は((クラスタ番号)-2)*(クラスタあたりのセクタ数)+(データ領域の開始セクタ番号)でセクタ番号に換算できます。

４．４　ブートセクタ

　ブートセクタはセクタ 0 に存在し、メディア情報を記録してある他、MSX-DOS の起動プログラムが書かれています。起動時にブートセクタが読み込まれ MSXDOS.SYS があればDOS の起動が行われます。MSX-DOS と MSX-DOS2 ではブートセクタの内容が異なります。

　メディアに固有の情報をブートセクタの内容から知ることができますが、逆にブートセクタのメディア情報を書き換えてメディアの構造を変えることはできません。DOS のアクセスではブートセクタの情報からではなく FAT に記録されているメディア ID(FAT ID)によってメディアの特性が決定されるからです。

ブートセクタの内容(DOS1)

```
+00H,3   8086 でのジャンプ命令(MS-DOS への配慮)
+03H,8   メーカー ID(メーカー名等)
+0BH,2   1 セクタのサイズ(バイト単位)
+0DH,1   1 クラスタのサイズ(セクタ単位)
+0EH,2   MSX-DOS が使用しないセクタ数
+10H,1   FAT の個数
+11H,2   ディレクトリエントリの数
+13H,2   ディスクに含まれるセクタの総数
+15H,1   メディア ID
+16H,2   FAT のサイズ(セクタ単位)
+18H,2   １トラックに含まれるセクタ数
+1AH,2   面の数
+1CH,2   隠されたセクタ数
+1EH     ブートプログラム
```

ブートセクタの内容(DOS2)

```
+00H～+1DH   DOS1 と同じ
+1EH,2   ブートプログラムの先頭(+30H へのジャンプ命令)
+20H,6   ”VOL_ID”という文字列
+26H,1   UNDEL 可能かどうかのフラグ(0 以外で可能)
+27H,4   ボリューム ID。フォーマット時に乱数により決定される。各バイトは 0～127。
+2BH,5   将来のための予約(0 で埋める)
+30H     ブートプログラム
```

４．５　FAT(File allocation table)

　FAT はデータ領域に記録されているデータのリンク情報を記録しています。データ領域は常に先頭から順に使用されていくとは限らず、飛び飛びにデータが記録される場合もあり、どのようにデータがつながっているかということがリンク情報として記録されます。FAT には 12 ビットのリンク情報が特殊なフォーマットで記録されています。リンク情報はそのクラスタに続くクラスタの番号を記録します。リンク情報が FFFH の場合そのクラスタがファイルの終端であることを、0 である場合そのクラスタが未使用であることを表します。
　FAT の先頭には FAT ID(メディア ID と同じ値)が記録されており、メディアの判別に使用されるため、ここが正しい値でないとファイル操作ができなくなります。

FAT の構造

```
+00H  FAT ID
+01H  0FFH
+02H  0FFH
+03H              FAT エントリ 2(下位 8 ビット)
+04H  FAT エントリ 3(下位 4 ビット)  FAT エントリ 2(上位 4 ビット)
+05H              FAT エントリ 3(上位 8 ビット)
+06H              FAT エントリ 4(下位 8 ビット)
+07H  FAT エントリ 5(下位 4 ビット)  FAT エントリ 4(上位 4 ビット)
+08H              FAT エントリ 5(上位 8 ビット)
+09H              FAT エントリ 6(下位 8 ビット)
+0AH  FAT エントリ 7(下位 4 ビット)  FAT エントリ 6(上位 4 ビット)
              :                 :                 :
  余った部分は 00H でうめられる。
```

４．６　ディレクトリ

　ディレクトリはディスクに記録されているファイルの情報を記録しています。ファイル毎に32バイトのディレクトリエントリが存在し、ここに記録されている情報は、ファイル名、ファイル属性、作成日時、ファイルの先頭クラスタ、ファイルサイズがあります。
　ファイル名として使用できる文字は、A～Z、0～9、$&#@!%'()-{}～_ひらがなカタカナです。アルファベットの小文字は、MSX-DOSによって大文字に変換されます。もしMSX-DOS以外によってMSX-DOSで使用できない文字(アルファベット小文字など)を使用したディレクトリエントリが作成されるとそのファイルに対してMSX-DOSではアクセスできません。
　DOS2においてサブディレクトリが作成された場合には、サブディレクトリに関する情報がディレクトリに記録されます。DOS1でのディレクトリはDOS2のルートディレクトリに当たります。サブディレクトリ本体はデータ領域に作成され、その領域の管理はファイルと同様にFATで行われます。サブディレクトリの先頭には"."と".."のサブディレクトリ属性を持ったディレクトリエントリが作成されます。これは、そのサブディレクトリその物と親ディレクトリを示します。
　ディレクトリエントリの先頭が00Hなら未使用のディレクトリエントリ、E5Hなら削除されたディレクトリエントリです。ファイル名の先頭がE5H(半角の"な"や全角文字の一部の1バイト目)の場合、その文字は05Hで記録されます。ファンクションコールではファイル名の先頭の文字がE5Hでも問題ありませんが、ディレクトリをセクタリードで直接読む場合は注意が必要です。
　DOS1ではファイル属性として、不可視があり、DOS2では更に、書き込み専用、読み込み専用、システムファイルの各属性があります。

ディレクトリエントリの構造

+00H,8　主ファイル名
+08H,3　拡張子
+0BH,1　ファイル属性
+0CH,1　(DOS2のみ)ファイル削除時にディレクトリエントリの第1文字が入る(UNDEL用)
+0DH,9　空き領域
+16H,2　ファイル更新時刻
+18H,2　ファイル更新日
+1AH,2　ファイルの先頭クラスタ
+1CH,4　ファイルサイズ

DOS2でボリューム名は+00H,11に記録されます。

ファイル属性

b0	読み出し専用(DOS2) ファイルの書き込みや削除ができなくなります。読み込み、ファイル名変更、移動は可能です。
b1	不可視 通常はファイルアクセスができなくなります。 DOS2で検索属性バイトの不可視をセットして「最初のエントリの検索」ファンクションを使用した時のみ見付けることができます。
b2	システムファイル(DOS2) 「新しいエントリの検索」と、「作成」ファンクションが自動的に削除することがないということ、を除いて不可視と同じ効果を持ちます。コマンドインタプリタによって組み込まれたコマンドはアクセスすることができません。
b3	ボリューム名(DOS2) ボリューム名を表します。ボリューム名はディレクトリエントリの最初の11バイトに書き込まれます。この属性はルートディレクトリに一つだけ設定可能です。ボリューム名にはコントロールコードと”/”を除いてファイル名として無効な文字も含めることができます。ただし、先頭に空白は入れられません。
b4	ディレクトリ(DOS2) サブディレクトリを表します。
b5	アーカイブビット(DOS2) ファイルが書き込まれてクローズされるとセットされます。このビットがセットされているとファイルが変更されたことを表します。

サブディレクトリ

　サブディレクトリを示すディレクトリエントリではファイルサイズは0で、先頭クラスタが、サブディレクトリの先頭クラスタを示しています。
　サブディレクトリ中には"."と".."という名前のディレクトリが作成されます。これらは、そのディレクトリ自身と親ディレクトリを示します。親ディレクトリがルートディレクトリだった場合、ディレクトリエントリ中の先頭クラスタ番号は0になります。ディレクトリエントリ"."と".."はディレクトリの先頭に作成されますが、必ずしも先頭にある必要はありません。

４．７　FCB(ファイルコントロールブロック)

FCB は、ファイルアクセスの際に必要となるファイル情報を保持する 37 バイトのテーブルです。オープンするファイル毎に存在します。

オープンされていない FCB とは、FCB のドライブ番号とファイル名及び拡張子をセットしたものです。オープンされていない FCB はファンクションコールによって各データがセットされオープンされた FCB となります。

FCB の内容は DOS1 と DOS2 では異なる部分があります。

FCB の内容(DOS1)

```
+00,1　ドライブ番号
+01,8　ファイル名前部
+09,3　ファイル名拡張子
+12,2　カレントブロック
+14,2　レコードサイズ
+16,4　ファイルサイズ
+20,2　日付
+22,2　時刻
+24,1　デバイス ID
+25,1　ディレクトリロケーション
+26,2　ファイル先頭のクラスタ番号
+28,2　最後にアクセスしたクラスタ番号
+30,2　ファイルの開始クラスタからの相対位置
+32,1　カレントレコード
+33,4　ランダムレコード
```

ドライブ番号

```
 0　　　カレントドライブ
 1～8　A:～H:
```

ファイル名前部、ファイル拡張子

文字数が満たない場合は空白で埋められる。

カレントブロック

シーケンシャルアクセスで参照中のブロック番号

レコードサイズ

レコードのサイズをバイト単位で指定

ファイルサイズ

バイト単位

日付、時刻

ディレクトリに記録されているのと同じフォーマット

デバイス ID

ディスクファイルの場合 40H+ドライブ番号
CON=FFH PRN=FBH LST=FCH AUX=FEH NUL=FDH

ディレクトリロケーション

ディレクトリ領域の中で何番目のディレクトリにエントリに該当するか

先頭クラスタ

ディスクにおける先頭クラスタを示す
最終アクセスクラスタ
最後にアクセスされたクラスタを示す
最終アクセスクラスタの先頭クラスタからの相対位置

カレントレコード

シーケンシャルアクセス時の、現在参照中のレコード

ランダムレコード(0～)

ランダムアクセス及びランダムブロックアクセスの際、アクセスするレコードを指定します。レコードサイズが 64 未満の場合 4 バイト、64 以上の場合は 3 バイトのみ有効です。

FCB の内容(DOS2)

+00,1　ドライブ番号
+01,8　ファイル名
+09,3　ファイル名拡張子
+12,1　エクステント番号(下位バイト)
+13,1　ファイル属性
+14,1　CP/M ファンクションのためのエクステント番号(上位バイト)
+15,1　CP/M ファンクションのレコードカウント
+16,4　ファイルサイズ(最下位バイトが最初にくる)
+20,4　ボリューム ID
+24,8　内部情報
+32,1　範囲(0～127)中のカレントレコード
+33,4　ランダムレコード番号(下位バイトが先)

+14,2 は DOS1 互換のブロックファンクションでは、レコードサイズをセットする。

４．８　DPB(ドライブパラメータブロック)

DPB は 21 バイトのテーブルで、論理ドライブ毎に存在しドライブ毎の情報を保持しています。この情報により MSX-DOS はメディア間の差異を吸収することができます。DPB の情報は MSX-DOS の起動時やメディアが交換された際に更新されます。

DPB の内容

```
+00,1　ドライブ番号
+01,1　メディア ID
+02,2　セクタサイズ
+04,1　ディレクトリマスク(セクタサイズ/32-1)
+05,1　ディレクトリシフト
+06,1　クラスタマスク(クラスタ当たりのセクタ数-1)
+07,1　クラスタシフト
+08,2　1 つめの FAT の先頭セクタ
+10,1　FAT の数
+11,1　ディレクトリエントリの数(～254)
+12,2　データ領域の先頭セクタ
+14,2　クラスタ総数+1
+16,1　1 つの FAT に要するセクタの数
+17,2　ディレクトリ領域の先頭セクタ
+19,2　メモリ上に置かれた FAT のアドレス
```

４．９　メディア ID

メディア ID はブートセクタや FAT の先頭に記録されていて、その値を読むことによりそのメディアを判別することができます。

メディア ID とメディアタイプの関係を示します。

```
F8H:1DD 9SEC   F9H:2DD 9SEC   FAH:1DD 8SEC   FBH:2DD 8SEC
FCH:1D  9SEC   FDH:2D  9SEC   FEH:1D  8SEC   FFH:2D  8SEC
```

RAM ディスクではメディア ID は FFH になっています。

4．10　FIB(ファイル情報ブロック)

　DOS2 で新設されたファイルに関係するファンクションでは、ASCIIZ 文字列の代わりに FIB を渡すことができます。FIB は未知のファイルやサブディレクトリを検索するといったより複雑な処理に使用されます。

　FIB は 64 バイトの領域で、「最初のエントリの検索」「次のエントリの検索」「新しいエントリの検索」ファンクションによって満たされます。

FIB の内容

+00,1　常に 0FFH(パス名文字列と区別するため)
+01,13　ファイル名(ASCIIZ 文字列)
+14,1　ファイル属性バイト
+15,2　最終変更時刻
+17,2　最終変更日付
+19,2　開始クラスタ
+21,4　ファイルのサイズ
+25,1　論理ドライブ
+26,38　内部情報(変更してはならない)

ファイル名

　直接印字可能な ASCIIZ 文字列です。空白は全て取り除かれ、あれば拡張子の前にピリオドが付加され、名前は大文字になります。エントリがボリュームラベルだと、名前は"."セパレータなしでストアされ、空白が残されて大文字化されません。

ファイル属性バイト

ビット	内容
b0	読み出し専用
b1	不可視
b2	システムファイル
b3	ボリューム名
b4	ディレクトリ
b5	アーカイブビット
b6	予約(常に 0)
b7	デバイスビット セットされると FIB がディスクファイルでなく、文字デバイスを参照していることを示し、他の全ての属性ビットは無視されます。

４．11　ASCIIZ 文字列

　DOS2 では ASCIIZ 文字列というものが多用されますが、ASCIIZ 文字列とはヌルコード(00H)で終了する文字列のことです。
　ドライブ・パス・ファイル ASCIIZ 文字列を受け取るファンクションで、ファンクションに渡すデータは、たとえば"B:\TEMP\ED.$$$"といった形の文字列の最後に 00H を付加したものになります。

４．12　時刻、日付のフォーマット

　ディレクトリエントリや FCB 中で時刻や日付を表す時、それぞれ２バイトに詰め込んだ特殊なフォーマットが使用されます。

時刻のフォーマット	日付のフォーマット
b15〜b11　時間(0〜23)	b15〜 b9　年(0〜99)(1980〜2079 に対応)
b10〜 b3　分(0〜59)	b8〜 b5　月(1〜12)
b4〜 b0　秒/2(0〜29)	b4〜 b0　日(1〜31)

4．13　ファイルハンドル

DOS2 では、新しいファンクションでファイルのアクセスを行う際、ファイルハンドルでファイルを指定します。

ファイルハンドルは特定のオープンファイルやデバイスを示す 8 ビットの数値です。ファイルハンドルは「ファイルハンドルのオープン」あるいは「ファイルハンドルの作成」ファンクションで割り当てられます。外部プログラムの実行時にはまえもっていくつかのファイルハンドルが定義されています。これらのファイルハンドルは自由にクローズすることができ、またプログラム終了の前に戻す必要はありません。

デフォルトのファイルハンドルを示します。

```
0  標準入力(CON)
1  標準出力(CON)
2  標準エラー出力(CON)
3  標準補助入出力(AUX)
4  標準プリンタ出力(PRN)
```

第５章　ファンクションコールの詳細

ファンクションコールの詳細を紹介します。

ファンクションの名称はMSX2テクニカルハンドブック、MSX-DOS2リファレンスマニュアル、MSXDatapack1、MSXDatapack3で相互に異なる場合がありますが、ここでは基本的にはDatapack3での表記に統一します。

ファンクションコール一覧

左から順に、CP/M2.2との互換性があるか、DOS1でのファンクションかどうか、DOS2でエラー処理ルーチンから呼び出せるか、DOS1とDOS2で注意すべき変更点の有無、ファンクション番号、パブリックラベル、ファンクションの名称、参照ページとなっています。

パブリックラベルはファンクションコードに割り当てられているラベルです。

CP/M	DOS1	ｴﾗｰ	変更		
●	●			00H _TERM0	プログラムの終了
●	●	●		01H _CONIN	コンソール入力
●	●	●		02H _CONOUT	コンソール出力
●	●	●		03H _AUXIN	補助入力
●	●	●		04H _AUXOUT	補助出力
●	●	●		05H _LSTOUT	プリンタ出力
●	●	●		06H _DIRIO	直接コンソール I/O
	●	●		07H _DIRIN	直接コンソール入力
	●	●		08H _INNOE	エコーなしコンソール入力
●	●	●		09H _STROUT	文字列出力
●	●	●		0AH _BUFIN	バッファ行入力
●	●	●		0BH _CONST	コンソールステータス
●	●	●		0CH _CPMVER	バージョン番号の獲得
●	●			0DH _DSKRST	ディスクリセット
●	●		●	0EH _SELDSK	ディスクの選択
●	●			0FH _FOPEN	ファイルのオープン[FCB]
●	●			10H _FCLOSE	ファイルのクローズ[FCB]
●	●			11H _SFIRST	最初のエントリの検索[FCB]
●	●		●	12H _SNEXT	次のエントリの検索[FCB]
●	●			13H _FDEL	ファイルの削除[FCB]
●	●		●	14H _RDSEQ	シーケンシャルな読み出し[FCB]
●	●			15H _WRSEQ	シーケンシャルな書き込み[FCB]
●	●			16H _FMAKE	ファイルの作成[FCB]
●	●			17H _FREN	ファイル名の変更[FCB]
●	●	●		18H _LOGIN	ログインベクタの獲得
●	●	●		19H _CURDRV	カレントドライブの獲得
●	●			1AH _SETDTA	ディスク転送アドレスのセット
	●		●	1BH _ALLOC	アロケーション情報の獲得
				1CH～20H	未定義
●	●			21H _RDRND	ランダムな読み出し[FCB]
●	●			22H _WRRND	ランダムな書き込み[FCB]
●	●			23H _FSIZE	ファイルサイズの獲得[FCB]
●	●			24H _SETRND	ランダムレコードのセット[FCB]

CP/M	DOS1	ｴﾗｰ	変更		
				25H	未定義
	●			26H _WRBLK	ランダムなブロックの書き込み[FCB]
	●			27H _RDBLK	ランダムなブロックの読み出し[FCB]
●	●			28H _WRZER	ゼロフィルを行うランダムな書き込み[FCB]
				29H	未定義
	●	●		2AH _GDATE	日付の獲得
	●	●		2BH _SDATE	日付のセット
	●	●		2CH _GTIME	時刻の獲得
	●	●		2DH _STIME	時刻のセット
	●	●		2EH _VERIFY	ベリファイフラグのセット・リセット
	●	●		2FH _RDABS	アブソリュートなセクタの読み出し
	●	●		30H _WRABS	アブソリュートなセクタの書き込み
		●		31H _DPARM	ディスクパラメータの獲得
				32H～3FH	未定義
				40H _FFIRST	最初のエントリの検索
				41H _FNEXT	次のエントリの検索
				42H _FNEW	新しいエントリの検索
				43H _OPEN	ファイルハンドルのオープン
				44H _CREATE	ファイルハンドルの作成
				45H _CLOSE	ファイルハンドルのクローズ
				46H _ENSURE	ファイルハンドルの確保
				47H _DUP	ファイルハンドルの複製
				48H _READ	ファイルハンドルからの読み出し
				49H _WRITE	ファイルハンドルへの書き込み
				4AH _SEEK	ファイルハンドルポインタの移動
				4BH _IOCTL	デバイスの I/O 制御
				4CH _HTEST	ファイルハンドルのテスト
				4DH _DELETE	ファイル・サブディレクトリの削除
				4EH _RENAME	ファイル名・サブディレクトリ名の変更
				4FH _MOVE	ファイル・サブディレクトリの移動
				50H _ATTR	ファイル属性の獲得・セット
				51H _FTIME	ファイルの日付および時刻の獲得・セット
				52H _HDELETE	ファイルハンドルの削除

CP/M	DOS1	ｴﾗｰ	変更		
				53H _HRENAME	ファイルハンドルの名前の変更
				54H _HMOVE	ファイルハンドルの移動
				55H _HATTR	ファイルハンドルの属性の獲得・セット
				56H _HFTIME	ファイルハンドルの日付及び時刻の獲得・セット
		●		57H _GETDTA	ディスク転送アドレスの獲得
		●		58H _GETVFY	ベリファイフラグ設定の獲得
				59H _GETCD	カレントディレクトリの獲得
				5AH _CHDIR	カレントディレクトリの変更
				5BH _PARSE	パス名の解析
				5CH _PFILE	ファイル名の解析
		●		5DH _CHKCHR	文字の検査
				5EH _WPATH	完全なパス文字列の獲得
				5FH _FLUSH	ディスクバッファのフラッシュ
				60H _FORK	子プロセスの起動
				61H _JOIN	親プロセスに戻る
				62H _TERM	エラーコードを伴った終了
				63H _DEFAB	アボート終了ルーチンの定義
				64H _DEFER	ディスクエラー処理ルーチンの定義
		●		65H _ERROR	直前のエラーコードの獲得
		●		66H _EXPLAIN	エラーコードの説明
				67H _FORMAT	ディスクのフォーマット
				68H _RAMD	RAM ディスクの作成あるいは消去
				69H _BUFFER	セクタバッファの割り付け
		●		6AH _ASSIGN	論理ドライブの割り当て
		●		6BH _GENV	環境変数の獲得
		●		6CH _SENV	環境変数のセット
		●		6DH _FENV	環境変数の検索
		●		6EH _DSKCHK	ディスク検査ステータスの獲得・セット
		●		6FH _DOSVER	MSX-DOS のバージョン番号の獲得
		●		70H _REDIR	リダイレクションの状態の獲得・セット

ファンクションの説明

　各ファンクションについて説明します。
それぞれのファンクションについて、ファンクション番号とパブリックラベル、ファンクションの名称、レジスタの設定と戻り値、説明があります。

00H _TERM0　プログラムの終了

設定　なし
戻り値　なし

MSX-DOS 上からコールした場合には、0000H に分岐することによってシステムがリセットされます(DOS のコマンドレベルに処理が返る)。Disk BASIC からコールした場合は、Disk BASIC がウォームスタートします。つまり、ロードされているプログラムを破壊せずに BASIC のコマンドレベルに戻ります。

DOS2 においては、プログラム終了の際にはこのファンクションコールではなく「エラーコードを返して終了」(62H)を使用することが推奨されています。

01H _CONIN　コンソール入力

設定　なし
戻り値　A(=L)　コンソールから入力した文字

入力がない場合(キーボードバッファが空の場合)には、入力待ちを行います。入力された文字はコンソールにエコーバックされます。以下に示すコントロールキャラクタはファンクション内部で処理されるため、入力として扱いません。

DOS2 においては、標準入力から読み込んで標準出力にエコーします。

CTRL-C　システムリセット
CTRL-P　プリンタへのエコー開始
CTRL-N　プリンタへのエコー停止
CTRL-S　ウェイト

02H _CONOUT コンソール出力

設定　E　出力する文字コード
戻り値　なし

　E レジスタで指定した文字を画面に表示します。またコンソールの入力をチェックし、CTRL+C、CTRL+P、CTRL+N、CTRL+S の処理を行います。
　DOS2 においては文字は標準出力に書き出されます。

03H _AUXIN　補助入力

　　設定　なし
戻り値　A(=L)　補助入力から読み込んだ１文字

文字がない場合は入力を待ちます。補助入力装置がセットされていない時は、EOF 文字(1AH)を返します。コンソール入力は、ファンクション 02H と同様にチェックします。

DOS2 においては補助入力デバイスから読み込まれます。

04H _AUXOUT 補助出力

　　設定　E　補助出力に出力する文字コード
　戻り値　なし

補助出力装置がセットされていない時は何もしません。コンソール入力はファンクション 02H と同様にチェックします。

DOS2 においては補助出力デバイスに対して出力されます。

05H _LSTOUT プリンタ出力

設定　E　プリンタに出力する文字コード
戻り値　なし

コンソール入力は、ファンクション 02H と同様にチェックします。
DOS2 においては標準プリンタデバイスに出力されます。

06H _DIRIO　直接コンソール I/O

```
設定　E　　FFH　　　入力
　　　　　　FFH以外　セットされた値を文字コードとしてコンソールに出力
戻り値　EがOFFHにセットされていた場合(入力)
　　　　　A(=L)　キーが押されていた場合はその文字コード
　　　　　　　　　押されていなかった場合は00H
　　　　EがOFFH以外にセットされていた場合(出力)
　　　　　なし
```

入力のエコーバック、コントロールキャラクタの処理は行いません。出力のプリンタへのエコーバックは行いません。

DOS2 においては入力は標準入力に対して、出力は標準出力に対して行われます。

07H _DIRIN　直接コンソール入力

設定　なし
戻り値　A(=L)　コンソールから入力した文字

コントロールキャラクタのサポートは行いません。エコーバックは行いません。DOS2 においては標準入力から読み込みます。

08H _INNOE　エコーなしコンソール入力

　　設定　なし
　戻り値　A(=L)　コンソールから入力した１文字

エコーバックは行いません。コントロールキャラクタはファンクション 01H と同様に処理します。

DOS2 においては標準入力から読み込みます。

09H _STROUT 文字列出力

　　設定　DE　コンソールに出力する文字列のアドレス
戻り値　なし

　文字列の最後には、終端記号(ターミネータ)として24H("$")を付加しておく必要があります。コントロールキャラクタはファンクション02Hと同様に処理します。

0AH _BUFIN　バッファ行入力

設定　DE　行バッファの先頭アドレス
　　　(行バッファの先頭+0)　最大入力文字数(1〜0FFH)
戻り値　(行バッファの先頭+1)　実際に入力された文字数
　　　(行バッファの先頭+2〜)　コンソールから入力された文字列

　リターンキーの入力をコンソールからの入力の終端と見なします。ただし、入力文字数が指定の最大入力文字数を超える場合には、指定の入力文字数までを入力文字列と見なしてメモリにセットした後、処理を終了します。このファンクションコールによる文字列入力時には、テンプレートによる編集が可能です。
　コントロールキャラクタの入力をチェックします。

0BH _CONST　コンソールステータス

設定	なし		
戻り値	A(=L)	FFH	コンソール入力がある
		00H	コンソール入力がない

入力文字がある時もその文字は取り出しません。ただし、コントロールキャラクタはファンクション 02H と同様に処理します。入力文字はファンクション 01H か 08H で取り出すことができます。

0CH _CPMVER バージョン番号の獲得

設定　なし
戻り値　HL(=BA)　0022H

エミュレートしている CP/M のバージョン番号を返します。一律に 0022H が返ります。

0DH _DSKRST ディスクリセット

設定　なし
戻り値　なし

変更されてまだディスクに書きこまれていないセクタがあれば、それをディスクに書き込んだ後、デフォルトドライブを A ドライブにセットし、DTA を 0080H にセットします。

0EH _SELDSK ディスクの選択

設定　E　デフォルトドライブ番号(0:A、1:B・・・7:H)
戻り値　(DOS1)なし
(DOS2)A(=L)　使用できるドライブの数

デフォルトドライブはFCBなどのドライブ番号に0が指定されたときに有効です。接続されていないドライブが指定された時には変更しません。

DOS2においてはCP/Mとの互換性のために(0004H)にもカレントドライブ番号を格納します。また使用できるドライブの数をA(1〜8)に返します。ただし、この数にRAMDISKは含まれません。

0FH _FOPEN　ファイルのオープン[FCB]

設定	DE	オープンされていないFCBの先頭アドレス	
戻り値	A(=L)	00H	成功
			FCBの各フィールドが設定される
		FFH	失敗

FCBがファイル入出力のファンクションコールで使用できるようになります。レコードサイズ、カレントレコード、ランダムレコードの各フィールドは使用するファンクションに応じて、オープンの後ユーザープログラムが設定しなければなりません。

10H _FCLOSE ファイルのクローズ[FCB]

```
  設定  DE   オープンされたFCBの先頭アドレス
戻り値  A(=L)   00H  クローズが成功
                FFH  クローズが失敗
```

現在の FCB の内容をディスク上の該当するディレクトリエリアに書きこむことによってファイルの更新に関する整合性を保ちます。ただし、ファイルの内容を変更していない時には書き込みは行いません。クローズした FCB はオープンされていない FCB として再利用できます。また、オープンされた FCB としてさらに入出力を続けることもできます。

11H _SFIRST 最初のエントリの検索[FCB]

設定　DE　オープンされていないFCBの先頭アドレス
戻り値　A(=L)　00H　ファイルが見つかった
DTAにドライブ番号と、それに続く32バイトにそのファイル
のディスク上のディレクトリエントリをセットする
FFH　ファイルが見つからなかった

ファイル名にはワイルドカード文字の「?」を使用することができます。その場合、最初にマッチしたファイルのディレクトリ情報を得ることができます。ワイルドカード文字の「*」はサポートしていないため、適当な数の「?」に置き換える必要があります。DTA領域に設定されたデータはオープンされていないFCBとして使用できます。

12H _SNEXT　次のエントリの検索[FCB]

　　設定　なし
　戻り値　A(=L)　00H　ファイルが見つかった
　　　　　　　　　FFH　ファイルが見つからない

「最初のエントリの検索」でのファイル名に適合する次のファイルを検索します。返される結果の内容は「最初のエントリの検索」と同様です。

DOS1 では「最初のエントリの検索」で指定した"オープンされていない FCB"が残っていなければなりません。DOS2 では情報を内部に保持するので"オープンされていない FCB"はなくても構いません。

13H _FDEL　ファイルの削除[FCB]

設定	DE	オープンされていないFCBのアドレス	
戻り値	A(=L)	00H	ファイルの削除が成功した場合
		FFH	ファイルが一つも削除されなかった場合

　ワイルドカードキャラクタが使用できます。
　DOS2においては、サブディレクトリ、システムファイル、不可視、読み出し専用ファイルは削除されません。

14H _RDSEQ　シーケンシャルな読み出し[FCB]

設定	DE	オープンされたFCBのアドレス	
戻り値	A(=L)	00H	読み出しが成功した
		01H	EOFでエラー

　128バイトのレコードを読み込みんでDTAに転送します。ファイルの終端で読み込んだバイト数が128バイトにみたない場合はDOS1では1AH、DOS2では00Hで詰められます。

15H _WRSEQ　シーケンシャルな書き込み[FCB]

　　設定　DE　オープンされたFCBのアドレス
　戻り値　A(=L)　00H　書き込みが成功した場合
　　　　　　　　　01H　ディスクがいっぱいでエラー

　128バイトのレコードをファイルに書き出します。レコードはカレントレコードで指定します。

16H _FMAKE　ファイルの作成[FCB]

設定　DE　オープンされていないFCBのアドレス
戻り値　A(=L)　00H　成功
　　　　　　　　FFH　失敗

指定のドライブのカレントディレクトリに新しいファイルを作成し、オープンします。ファイル名にワイルドカードを使用することはできません。また、不正なファイル名が生成されないようにチェックされます。

17H _FREN　ファイル名の変更[FCB]

```
  設定  DE  オープンされていないFCBのアドレス
戻り値  A(=L)  00H  成功
               FFH  失敗
```

　オープンされていないFCBは通常のドライブとファイル名に加えて、(DE+17)からに新しいファイル名をセットします。2番目のファイル名中、"?"についてはそれに対応する元の文字をそのままにします。ファイル名の重複や不正なファイル名をチェックされます。サブディレクトリやシステムファイル、不可視属性ファイルの名前変更はできません。

18H _LOGIN　ログインベクタの獲得

設定　なし
戻り値　HL　ログインベクタ

利用できるドライブについてHLの対応するビットを1にして返します。現在システムでサポートされているのは8つまでで、Hレジスタは通常0が返ります。

L　/ H: / G: / F: / E: / D: / C: / B: / A: /

19H _CURDRV カレントドライブの獲得

設定　なし
戻り値　A(=L)　カレントドライブ(0:A、1:B・・・7:H)

カレントドライブ番号を返します。

1AH _SETDTA ディスク転送アドレスのセット

設定　DE　要求するディスク転送アドレス
戻り値　なし

DE で指定したアドレスをディスク転送アドレス(DTA)としてセットします。このアドレスは FCB を利用したディスクアクセスなどで使用されます。
DOS2 で拡張されたファンクションにおいては、転送アドレスを直接指定するためにディスク転送アドレスは使用されません。

1BH _ALLOC　アロケーション情報の獲得

設定	E	ドライブ番号(0:カレント、1:A・・・8:H)
戻り値	A	1クラスタ当たりのセクタ数
	BC	セクタサイズ(常に512)
	DE	ディスク上のクラスタの総数
	HL	ディスク上の未使用クラスタ数
	IX	DPBへのポインタ
	IY	FATセクタへのポインタ

指定のドライブ中のディスクについての情報を返します。

DOS2ではDOS1と異なり、FATの最初のセクタだけがIYからのアドレスでアクセスでき、そこのデータは次のファンクションコールまでしか有効ではありません。

21H _RDRND　ランダムな読み出し[FCB]

設定	DE	オープンされたFCBへのポインタ	
戻り値	A(=L)	00H	読み出しが成功の場合
		01H	エンドオブファイルでエラー

ファイルから128バイトのレコードを読み出しDTAに入れます。ファイルの位置はFCBのランダムレコード番号で決まります。ファイルの終わりの断片レコードは0で詰められます。ランダムレコード番号は変更されません。

22H _WRRND　ランダムな書き込み[FCB]

　　設定　DE　オープンされた FCB のアドレス
　戻り値　A(=L)　00H　エラーが起きなかった場合
　　　　　　　　01H　ディスクがいっぱいでエラーの場合

　DTA から 128 バイトをファイルに書き込みます。書き込む位置はランダムレコード番号によって指定します。ランダムレコード番号は更新されません。

23H _FSIZE　ファイルサイズの獲得[FCB]

設定　DE　オープンされていないFCBのアドレス
戻り値　A(=L)　00H　ファイルが見つかった
　　　　　　　FFH　ファイルが見つからない

　ファイルのオープンと同様に最初に一致するファイルを検索します。ファイルのサイズは128バイト単位に切り上げられ、レコード数がランダムレコード番号の始めの3バイトにセットされます。

24H _SETRND ランダムレコードのセット[FCB]

設定　DE　オープンされたFCBのアドレス
戻り値　なし

FCB中のランダムレコードフィールドをカレントレコードによって設定します。

26H _WRBLK　ランダムなブロックの書き込み[FCB]

```
  設定  DE  オープンされたFCBのアドレス
        HL  書き込むレコード数
戻り値  A  00H  成功
           01H  エラー
```

DTA からのデータをランダムレコード番号によって決まる位置へ書き込みます。FCB のレコードサイズによってレコードのサイズが決まります。

ランダムレコードが更新されます。

書き込むレコード数が 0 の場合、データは書き込まれず、ファイルサイズがランダムレコード領域で指定した値に変更されます。このとき、必要に応じてディスクの領域が割り当てられたり開放されたりします。あらたに割り当てられた領域は初期化されません。

27H _RDBLK　ランダムなブロックの読み出し[FCB]

設定	DE	オープンされたFCBのアドレス	
	HL	読み出すレコード数	
戻り値	A	00H	成功
		01H	エラー（通常、エンドオブファイル）
	HL	実際に読んだレコード数	

EOFに達した場合、断片レコードは余りが0で詰められ、戻り値はAに01Hがセットされます。

28H _WRZER　ゼロフィルを行うランダムな書き込み[FCB]

```
　　設定　DE　オープンされたFCBのアドレス
　戻り値　A(=L)　00H　成功
　　　　　　　　 01H　エラー
```

　ランダムな書き込みと同じですが、ファイルを拡張した時にあらたに割り当てられたクラスタが0でうめられます。

2AH _GDATE　日付の獲得

設定	なし	
戻り値	HL	年(1980～2079)
	D	月(1～12)
	E	日(1～31)
	A	曜日(0:日～6:土)

2BH _SDATE　日付のセット

```
　設定　HL　年(1980～2079)
　　　　D　 月(1～12)
　　　　E　 日(1～31)
戻り値　A　00H　成功
　　　　　 FFH　日付が無効
```

曜日は日付から算出されます。

2CH _GTIME　時刻の獲得

設定	なし	
戻り値	H	時間(0～23)
	L	分(0～59)
	D	秒(0～59)
	E	1/100 秒(常に 0)

クロックが 1/100 秒単位までないため E は常に 0 です。

2DH _STIME　時刻のセット

設定	H	時間(0～23)
	L	分(0～59)
	D	秒(0～59)
	E	1/100秒(無視されます)
戻り値	A	00H　成功
		FFH　時刻が無効

2EH _VERIFY ベリファイフラグのセット・リセット

　　設定　E　0　　　ベリファイを無効に
　　　　　　0以外　ベリファイを有効に
　戻り値　なし

MSX-DOS が起動した時には無効になっています。
この機能はディスクドライバに依存しているため、ディスクドライバがベリファイに対応していない場合、このファンクションでベリファイを有効にしてもベリファイは行われません。

2FH _RDABS　アブソリュートなセクタの読み出し

設定	DE	セクタ番号
	L	ドライブ番号(0:A、1:B・・・7:H)
	H	読み出すセクタ数
戻り値	A	エラーコード(0でエラーなし)

データはDTAに読み込まれます。

30H _WRABS　アブソリュートなセクタの書き込み

```
   設定  DE  セクタ番号
         L   ドライブ番号(0:A、1:B・・・7:H)
         H   読み出すセクタ数
 戻り値  A   エラーコード(0でエラーなし)
```

データはDTAから書き込まれます。

31H _DPARM　ディスクパラメータの獲得

設定	DE	ディスクパラメータ(32バイトのバッファ)へのポインタ
	L	ドライブ番号(0:カレント、1:A・・・8:H)
戻り値	A	エラーコード
	DE	保存される

指定のドライブ中のディスクのフォーマットに関するパラメータを、ユーザープログラム内に確保されたバッファに返します。

ディスクパラメータの内容

```
+00,1  物理的なドライブ番号(1:A･･･8:H)
+01,2  セクタサイズ(現在は常に512)
+03,1  クラスタごとのセクタ数
+04,2  予約セクタ数(通常1)
+06,1  FATの数(通常2)
+07,2  ルートディレクトリのエントリ数
+09,2  論理セクタの総数
+11,1  メディアディスクリプタバイト
+12,1  FATごとのセクタ数
+13,2  ルートディレクトリの最初のセクタ番号
+15,2  最初のデータのセクタ番号
+17,2  最大クラスタ番号
+19,1  ダーティディスクフラグ
+20,4  ボリュームID(-1:ボリュームIDなし)
+24,8  システム予約(現在は常に0)
```

ダーティディスクフラグは、ディスク中にUNDELコマンドで復活できるファイルがあることを示すフラグです。ファイルやディレクトリにクラスタが割り当てられるとリセットされます。

40H _FFIRST 最初のエントリの検索

設定	DE	ドライブ・パス・ファイル ASCIIZ 文字列または FIB ポインタ
	HL	検索ファイル名 ASCIIZ 文字列(DE=FIB ポインタである場合)
	B	検索属性
	IX	新しい FIB へのポインタ
戻り値	A	エラーコード
	(IX)	一致するエントリが入る

文字列のドライブ・パス部分あるいは FIB は検索するディレクトリを指定します。ファイルを指定する FIB が渡されると.IATTR エラーが返されます。

検索するファイル名には、ワイルドカード文字("?"と"*")を含めても良く、その場合には最初に一致するエントリが返されます。ファイル名がヌルの場合"*.*"と見なします。

検索属性は一致するエントリのタイプを指定します。読み出し専用およびアーカイブビットは無視されます。ボリューム名ビットがセットされるとボリュームラベルのエントリだけが検索されます。ボリューム名はパスの指定は無視してルートから検索します。

DE が FIB を指している場合、IX も同一の FIB を指定することができ、この場合重ね書きされます。

41H _FNEXT　次のエントリの検索

設定	IX	以前の「最初のエントリの検索」で返されたFIBへのポインタ
戻り値	A	エラー
	(IX)	次の一致するエントリが入る

「最初のエントリの検索」の後で使用しなければなりません。
一致するエントリが見つからない場合、.NOFIL エラーを返します。

42H _FNEW　新しいエントリの検索

設定	DE	ドライブ・パス・ファイルASCIIZ文字列またはFIBポインタ
	HL	検索ファイル名ASCIIZ文字列(DE=FIBポインタの場合のみ)
	B	検索属性／新規作成フラグ(b7)
	IX	テンプレートファイル名を保持している新しいFIBへのポインタ
戻り値	A	エラー
	(IX)	新しいエントリが入る

「最初のエントリの検索」と似ていますが、指定されたファイル名で新しいエントリを作成します。ファイル名にワイルドカード文字を使用すると、テンプレートファイル名の該当する文字で置き換えられます。ファイル名がヌルの場合"*.*"として扱われます。

このファンクションコールは、ファイル名を変えてコピーを行う場合に有用です。

検索属性をセットすると該当するボリューム名、サブディレクトリ、などが作成されます。ファイルに不可視等の各属性をセットすることもできます。アーカイブビットは常にセットされて作成されます。

新規作成フラグをセットしてあると、既に該当するエントリが存在している場合は.FILEXエラーを返します。

新規作成フラグがセットされていない場合に、読み出し専用ファイル(.FILRO)、システムファイル(.SYSX)又はサブディレクトリ(.DIRX)だった場合やエントリをオープンしているハンドルが存在した場合(.FOPEN)にはエラーが返されます。

サブディレクトリを作成しようとしている場合、既存のファイルは削除されません。(.FILEXエラー)

.FILEX、.FILRO、.SYSX、.DIRX、.FOPENの各エラーが発生した場合、既存のエントリの情報がFIBに書き込まれます。

43H _OPEN　ファイルハンドルのオープン

設定　DE　ドライブ・パス・ファイル ASCIIZ 文字列または FIB ポインタ
　　　A　オープンモード
　　　　b0　書き込み禁止
　　　　b1　読み出し禁止
　　　　b2　継承
　　　　b3～b7　必ずクリア
戻り値　A　エラー
　　　B　新しいファイルハンドル

　読み出し禁止の場合、ファイルハンドルへの書き込みも禁止されます。継承ビットがセットされると「子プロセスの起動」によって生成された新しいプロセスにファイルハンドルが受け継がれます。

　デバイスのファイルハンドルが組み込みデバイスのファイル名を与えられてオープンされると、ASCII モードでオープンされます。

44H _CREATE ファイルハンドルの作成

設定　DE　ドライブ・パス・ファイル ASCIIZ 文字列
　　　A　 オープンモード
　　　　b0　書き込み禁止
　　　　b1　読み出し禁止
　　　　b2　継承
　　　　b3～b7　必ずクリア
　　　B　要求する属性／新規作成フラグ(b7)
戻り値　A　エラー
　　　B　新しいファイルハンドル

ファイルまたはサブディレクトリを作成します。

新規作成フラグをセットすると既に同名のエントリが存在している場合は、既存のファイルを削除せずエラーを返します。

不正な属性ビットは無視されます。ファイルは常にアーカイブビットをセットして作成されます。

ファイルはオープンされてファイルハンドルが返されます。(「ファイルハンドルのオープン」と同様)

サブディレクトリはオープンされないため、B レジスタに FFH が返されます。

45H _CLOSE　ファイルハンドルのクローズ

設定　B　ファイルハンドル
戻り値　A　エラー

　指定のファイルハンドルを開放して再利用できるようにします。ハンドルにファイルが書き込まれていると、ディレクトリエントリを更新し、アーカイブビットをセットし、バッファ中のデータをディスクの書き出します。
　このファイルハンドルは続けて使用することはできず、「ファイルハンドルの複製」や「子プロセスの起動」で作成されたファイルハンドルは続けて使用できます。

46H _ENSURE ファイルハンドルの確保

設定　B　ファイルハンドル
戻り値　A　エラー

ファイルハンドルにファイルが書き込まれていると、ディレクトリエントリを更新し、アーカイブビットをセットし、バッファ中のデータをディスクに書き出します。ファイルハンドルは開放されないため、続けてファイルをアクセスすることができます。ファイルポインタの位置は変わりません。

47H _DUP　ファイルハンドルの複製

設定	B	ファイルハンドル
戻り値	A	エラー
	B	新しいファイルハンドル

ファイルハンドルのコピーを作成します。

元のファイルハンドルと新しいファイルハンドルの両方を同様に使用することができます。ファイルポインタの移動や名前の変更は両方のファイルハンドルに効果が及びます。ただし、片方のファイルハンドルを削除した場合、もう片方のファイルハンドルはオープンしたままの状態ですが「クローズ」、「確保」、「削除」以外では使用できません。

48H _READ　ファイルハンドルからの読み出し

設定	B	ファイルハンドル
	DE	バッファアドレス
	HL	読み込むバイト数
戻り値	A	エラー
	HL	実際に読み込んだバイト数

指定されたバイト数をファイルポインタの位置から読み出します。読み出した後、ファイルポインタは続くアドレスに更新されます。

HL に読み込んだバイト数が返りますが、指定のバイト数よりも短い場合もあります。ただし、指定のバイト数よりも短い場合でも.EOF エラーは返りません。.EOF エラーは、実際には次に読み込みをした時に 0 バイトの読み込みを行って返ります。

49H _WRITE　ファイルハンドルへの書き込み

設定	B	ファイルハンドル
	DE	バッファアドレス
	HL	書き込むバイト数
戻り値	A	エラー
	HL	実際に書き込んだバイト数

指定されたバイト数をファイルポインタの位置から書き込みます。書き込んだ後、ファイルポインタは続くアドレスに更新されます。

実際に書き込んだバイト数は、エラーが発生した時に 0 になる他は要求した数と等しく通常は無視できます。

4AH _SEEK　ファイルハンドルポインタの移動

設定	B	ファイルハンドル
	A	方式コード
		0　ファイルの先頭から相対
		1　現在の位置から相対
		2　エンドオブファイルから相対
	DE:HL	符号付きオフセット
戻り値	A	エラー
	DE:HL	新しいファイルポインタ

ファイルポインタを移動します。

エンドオブファイルのチェックは行わないので、ファイルポインタはエンドオブファイルを越えてセットすることが可能です。

4BH _IOCTL　デバイスの I/O 制御

設定　B　ファイルハンドル
　　　A　サブファンクションコード
　　　　　00H　ファイルハンドルの状態の獲得
　　　　　01H　ASCII・バイナリモードのセット
　　　　　02H　入力レディの検査
　　　　　03H　出力レディの検査
　　　　　04H　画面サイズの検出
　　　DE　他のパラメータ
戻り値　A　エラー
　　　DE　他の結果

ファイルハンドルの種々の状態を判別、変更することができます。特にディスクファイルとデバイスを区別することができ、それらを区別したい場合に役に立ちます。どの処理を行うかはサブファンクションコードで指定します。
　サブファンクションで指定される各機能を以下に説明します。

ファイルハンドルの状態の獲得(A=0)

戻り値　DE　デバイスの場合
　　b0　コンソール入力デバイス
　　b1　コンソール出力デバイス
　　b2～b4　システム予約
　　b5　ASCII モード
　　b6　エンドオブファイル
　　b7　デバイス(常に 1:デバイスファイル)
　　b8～b15　システム予約
　ディスクファイルの場合
　　b0～b5　ドライブ番号(0:A・・・)
　　b6　エンドオブファイル
　　b7　デバイス(常に 0:ディスクファイル)
　　b8～b15　システム予約

ASCII・バイナリモードのセット(A=1)

```
設定  E  b5  1  ASCII
             0  バイナリ
```

DE レジスタの他のビットは無視されます。

入力レディ(A=2)／出力レディの検査(A=3)

```
戻り値  E  FFH  レディ
           00H  ノットレディ
```

画面サイズの獲得(A=4)

```
戻り値  D  行数
        E  列数
```

画面サイズのないデバイスの場合 D も E も 0 になります。どちらが 0 でもそれは無限であると解釈します。

4CH _HTEST　ファイルハンドルのテスト

設定	B	ファイルハンドル	
	DE	ドライブ・パス・ファイル ASCIIZ 文字列または FIB ポインタ	
戻り値	A	エラー	
	B	00H	同じファイルでない
		FFH	同じファイル

ファイルハンドルと DE で指定されるファイルが同じファイルかどうかを判断します。

ファイルがデバイスファイルの場合は同じファイルでないと常に返します。

4DH _DELETE ファイル・サブディレクトリの削除

設定 DE ドライブ・パス・ファイルASCIIZ文字列またはFIBポインタ
戻り値 A エラー

ファイルあるいはサブディレクトリを削除します。ワイルドカード文字は使用できません。つまり、一つだけのファイルあるいはサブディレクトリを削除できます。

サブディレクトリは、空でない場合には削除できません(.DIRNE)。オープンされているファイル(.FOPEN)や読み出し専用ファイル(.FILRO)は削除できません。

FIBを渡した場合、FIBの指すファイルは削除されてしまうため、そのFIBは「次のエントリの検索」以外では使用してはいけません。

「CON」などのデバイス名が指定されるとエラーは返されませんが、そのデバイスは実際には削除されません。

4EH _RENAME ファイル名・サブディレクトリ名の変更

設定　DE　ドライブ・パス・ファイルASCIIZ文字列またはFIBポインタ
　　　HL　新しいファイル名ASCIIZ文字列
戻り値　A　エラー

ファイル名にはワイルドカード文字を使用することができます。新しいファイル名にワイルドカード文字を使用すると、変更前のファイル名のその位置に該当する文字で置き換えられます。

DEがFIBを指している場合、FIB自身に含まれるファイル名の情報は更新されないため、FIBをそのまま使用することはできません。

新しいファイル名文字列にドライブ文字やディレクトリパスを含めると、.IFNMエラーが返されます。新しいファイル名が不正である場合(.IFNM)やすでに同名のエントリが存在する場合(.DUPF)エラーになります。「CON」などのデバイス名が指定されるとエラーは返されませんが、名前の変更はされません。

4FH _MOVE　ファイル・サブディレクトリの移動

設定	DE	ドライブ・パス・ファイル ASCIIZ 文字列または FIB ポインタ
	HL	新しいパス ASCIIZ 文字列
戻り値	A	エラー

新しいパス文字列中にはドライブ名があってはなりません。

ワイルドカード文字を使用することはできません。

ディレクトリは自分自身の下に移動することはできません(.DIRE)。またオープンされたファイルハンドルを持つファイルは移動できません(.FOPEN)。

DE が FIB を指している場合、FIB 自身に含まれるファイルの位置に関する情報は更新されないため、この FIB をそのまま使用することはできません。「次のエントリの検索」では使用することができます。

50H _ATTR　ファイル属性の獲得・セット

設定　DE　ドライブ・パス・ファイルASCIIZ文字列またはFIBポインタ
　　　A　0　属性の獲得
　　　　　1　属性のセット
　　　L　新しい属性バイト(A=1の場合)
戻り値　A　エラー
　　　L　現在の属性バイト

ファイルについては、システム、不可視、読み出し専用、アーカイブビットを変更することができます。サブディレクトリについては不可視属性を変更できます

FIBが渡された場合、FIB中の属性バイトは更新されません。

ワイルドカードは使用できません。オープンされているファイルの属性を変更することはできません。"."と".."の属性は変更することができます。

「CON」などのデバイス名が渡されてもエラーは返されませんが、実際には属性は変更されません。

51H _FTIME　ファイルの日付および時刻の獲得・セット

設定	DE	ドライブ・パス・ファイル ASCIIZ 文字列または FIB ポインタ
	A	0　日付と時刻の獲得
		1　日付と時刻のセット
	IX	新しい時刻の値(A=1 の場合)
	HL	新しい日付の値(A=1 の場合)
戻り値	A	エラー
	DE	ファイルの時刻の現在値
	HL	ファイルの日付の現在値

ファイルまたはディレクトリの最終修正時刻を獲得またはセットします。

ワイルドカードは使用できません。オープンしているファイルでは変更できません(読み出しは可)。日付と時間のチェックは行われません。

FIB を渡した場合 FIB 中の日付と時刻は更新されません。

52H _HDELETE ファイルハンドルの削除

　設定　B　ファイルハンドル
戻り値　A　エラー

　指定したファイルを削除しそのファイルハンドルをクローズします。別個に同じファイルをオープンしているファイルハンドルが存在すると削除は行えません(.FOPEN)。
ファイルハンドルの複製が存在していた場合、これらの複製は使用できなくなり、使用しようとした場合.HDEAD エラーとなります。
　エラーが存在しても常にファイルハンドルはクローズされます。

53H _HRENAME ファイルハンドルの名前の変更

設定　B　ファイルハンドル
　　　HL　新しいファイル名ASCIIZ文字列
戻り値　A　エラー

ファイルハンドルに結合されたファイルの名前を変更します。同一のファイルに対して別にオープンされたファイルハンドルが存在する場合名前の変更はできません。

ファイルハンドルのコピーが存在する場合は、コピーも同時に名前が変更されます。

ファイルハンドルの名前の変更はファイルポインタを変えませんが、暗黙の「確保」処理を実行します。

54H _HMOVE　ファイルハンドルの移動

　　設定　B　　ファイルハンドル
　　　　　HL　新しいパス ASCIIZ 文字列
　　戻り値　A　エラー

　指定のファイルハンドルに結合したファイルを移動します。
　別にオープンしているファイルハンドルが存在する場合は移動できません(.FOPEN)。
　ファイルハンドルのコピーが存在する場合、コピーも移動されます。ファイルハンドルの移動はファイルポインタを変更しませんが、暗黙の「確保」処理をします。

55H _HATTR　ファイルハンドルの属性の獲得・セット

```
   設定  B  ファイルハンドル
         A  0  属性の獲得
            1  属性のセット
         L  新しい属性バイト(A=1の場合)
 戻り値  A  エラー
         L  現在の属性バイト
```

　指定のファイルハンドルに結合したファイルの属性バイトを獲得または変更します。

　別にオープンしているファイルハンドルが存在する場合は獲得はできますが、変更はできません(.FOPEN)。

　ファイルポインタを変更しませんが、暗黙の「確保」処理をします。

56H _HFTIME ファイルハンドルの日付及び時刻の獲得・セット

設定	B	ファイルハンドル	
	A	0	日付と時刻の獲得
		1	日付と時刻のセット
	IX	新しい時刻の値(A=1 の場合)	
	HL	新しい日付の値(A=1 の場合)	
戻り値	A	エラー	
	DE	ファイルの時刻の現在値	
	HL	ファイルの日付の現在値	

指定のファイルハンドルに結合したファイルの日付と時刻を獲得または変更します。

別にオープンしているファイルハンドルが存在する場合は獲得はできますが、変更はできません(.FOPEN)。

ファイルポインタを変更しませんが、暗黙の「確保」処理をします。

57H _GETDTA ディスク転送アドレスの獲得

設定　なし
戻り値　DE　現在のディスク転送アドレス

58H _GETVFY ベリファイフラグ設定の獲得

設定　なし
戻り値　B　00H　リファイ無効
　　　　　　FFH　ベリファイ有効

59H _GETCD　カレントディレクトリの獲得

設定　B　ドライブ番号(0:カレント、1:A・・・8:H)
　　　DE　64バイトバッファへのポインタ
戻り値　A　エラー
　　　DE　カレントパスで満たされる

　指定のドライブのカレントディレクトリを表すASCIIZ文字列をバッファに格納します。文字列にはドライブ名や前後の"\"は含まれません。ルートはヌル文字列となります。ドライブがアクセスされて、カレントディレクトリが実際に現在のディスク上に存在することが確認されるので、存在しない場合にはカレントディレクトリはルートにセットされヌル文字列が返されます。

5AH _CHDIR　カレントディレクトリの変更

設定　DE　ドライブ・パス・ファイル ASCIIZ 文字列
戻り値　A　エラー

ドライブ・パス・ファイル文字列はファイルではなくディレクトリを指定しなければなりません。指定のディレクトリが存在しない場合、ディレクトリは変更されずエラーが返ります(.NODIR)。

5BH _PARSE　パス名の解析

設定	B	ボリューム名フラグ(b4)
	DE	解析するASCIIZ文字列
戻り値	A	エラー
	DE	終了文字へのポインタ
	HL	最後の項目の先頭へのポインタ
	B	解析フラグ
	C	論理ドライブ番号(1:A、2:B・・・8:H)

このファンクションは純粋な文字列操作ファンクションでディスクをアクセスすることはなく、ユーザーの文字列に変更を加えることもありません。外部プログラムがコマンド行を解析する手助けのために用意されています。

ボリューム名フラグがリセットされていると文字列をドライブ・パス・ファイル文字列として、セットされていると文字列をドライブ・ボリューム文字列として解析します。

DEに返されるポインタはパス名文字列中で有効でない最初の文字を指し、文字列の終わりのヌルのこともあります。HLに返されるポインタは文字列の最後の項目(ファイル名部分)の最初の文字を指します。

Cに返されるドライブ番号は文字列中で指定される論理ドライブです。文字列がドライブ文字で始まらない時は、カレントドライブが暗黙で指定されたということなので、Cはカレントドライブ番号を保持しています。Cには0が返されることはありません。

Bに返される解析フラグは文字列に関する情報を示します。ボリューム名ではビット1、4、5、6、7は常にクリアされます。

解析フラグ

b0　ドライブ名以外の文字が解析された
b1　ディレクトリパスが指定された
b2　ドライブ名が指定された
b3　最後の項目で主ファイル名が指定された
b4　最後の項目でファイル名拡張子が指定された
b5　最後の項目にワイルドカードがある
b6　最後の項目が”.”か”..”の時セット
b7　最後の項目が”..”の時セット

5CH _PFILE　ファイル名の解析

設定	DE	解析のためのASCIIZ文字列
	HL	11バイトバッファへのポインタ
戻り値	A	エラー
	DE	終了文字へのポインタ
	HL	保存、バッファが使用
	B	解析フラグ

このファンクションは純粋な文字列操作ファンクションでディスクをアクセスすることはなく、ユーザーの文字列に変更を加えることもありません。外部プログラムがフォーマットされた形式でファイル名を出力する手助けのために用意されています。

ASCIIZ文字列は単一のファイル名の項目として解析され、ファイル名はユーザーの11バイトのバッファに拡張された形式でセットされ、ファイル名及び拡張子の余白には空白が詰められます。

Bに返される解析フラグは「パス名の解析」と同じですが、ビット0〜2は常にクリアされます。文字列中に有効なファイル名がなかったとしてもバッファは満たされ、その場合にはバッファは空白で満たされます。"*"は適当な数の"?"に展開されます。ファイル名やファイル名拡張子が長すぎると、余分な文字は無視されます。

DEに返されるポインタはファイル名の一部ではなかった文字列中の最初の文字を指しますが、文字列の終わりにあるヌルである場合があります。DEが指す文字は有効なファイル名の文字であることはありません。

5DH _CHKCHR 文字の検査

設定	D	文字フラグ
	E	検査する文字
戻り値	A	0(エラーを返すことはない)
	D	変更された文字フラグ
	E	検査された(大文字にされた)文字

言語設定と独立して文字を大文字にしたり2バイトの文字やファイル名の操作をしたりするのに役立ちます。

文字フラグ

b0　大文字にしない
b1　2バイト文字の第1バイト
b2　2バイト文字の第2バイト
b3　ボリューム名の文字(ファイル名でない)
b4　ファイル・ボリューム名の文字
b5～7　システム予約(常にクリア)

ビット1～2は文字列の最初の文字をチェックする時に両方ともクリアすることができ、返された設定は続く文字のためにこのファンクションに直接渡すことができます。これらのフラグで2バイト文字を含む可能性のある文字列を逆戻りする場合は注意が必要です。

ビット4は文字がファイル名やボリューム名の終了文字であるとセットされてリターンします。ビット3は単にファイル名の検査かあるいはボリューム名の文字の検査かを決定するのに使われます。2バイト文字はボリューム名あるいはファイル名の終了文字と見なされることはありません。

5EH _WPATH　完全なパス文字列の獲得

設定	DE	64バイトバッファへのポインタ
戻り値	A	エラー
	DE	完全なパス文字列で満たされる
	HL	最後の項目の始めへのポインタ

内部バッファから ASCIIZ パス文字列をユーザーのバッファへコピーします。文字列は、前に実行された「最初のエントリの検索」あるいは「新しいエントリの検索」で見付けられたファイルやサブディレクトリのルートディレクトリからの完全なパスとファイル名を表します。文字列はドライブや最初の"\"を含みません。HL レジスタは「パス名の解析」と同様、文字列上の最後の項目の最初の文字を指しています。

「最初のエントリの検索」や「新しいエントリの検索」で DE が ASCIIZ 文字列を指して実行された場合、続く「完全なパス名の獲得」は検索で返される FIB に関係するサブディレクトリやファイルを表す文字列を返します。これがサブディレクトリであると FIB はレジスタ DE で別の最初のエントリの検索に渡すことができます。

多くのファンクションコールが内部のパス文字列を変更し、たいていの場合正しくないので、このファンクションを使用する時は注意が必要です。「完全なパスの獲得」は「最初のエントリの検索」や「新しいエントリの検索」の直後に行うべきです。

5FH _FLUSH　ディスクバッファのフラッシュ

```
設定  B  ドライブ番号(0:カレント、1:A・・・、0FFH:全て)
      D  00H  フラッシュのみ
         FFH  フラッシュして無効にする
戻り値 A  エラー
```

指定のドライブについて、まだ書き出されていない全てのディスクバッファをフラッシュします。D が FFH であるとそのドライブの全てのバッファも無効になります。

60H _FORK　子プロセスの起動

設定　なし
戻り値　A　エラー
　　　　B　親プロセスのプロセス ID

システムに対して子プロセスが起動されようとしていることを報告します。一般的にはこれは実行されようとする新しいプログラムやサブコマンドです。

新しいファイルハンドルのセットが作成され、継承アクセスモードをセットしてオープンされた全ての現在のファイルハンドルがコピーされます。標準ファイルハンドルは継承できるためコピーされます。

新しいプロセス ID が子プロセスのために割り当てられ、親プロセスのプロセス ID が返され、後で「親プロセスに戻る」で親プロセスに戻ることができます。
ファイルハンドルを作成するのにメモリが足りない場合はエラーが返されます。

子プロセスはオリジナルではなくて以前のファイルハンドルのコピーを持つため、コピーの一つがクローズされても元の物はオープンされたままです。従って、子プロセスが標準出力のファイルハンドルをクローズしそれを新しいファイルに再オープンすると、親プロセスに戻るが実行されて親プロセスに戻った時、元の標準出力チャンネルはまだ有効です。

61H _JOIN　親プロセスに戻る

設定	B	親のプロセスIDまたは0
戻り値	A	エラー
	B	子プロセスからの1次エラーコード
	C	子プロセスからの2次エラーコード

指定の親プロセスに戻り、子プロセスが終了したエラーコードをBに入れ、子プロセスからの2次エラーコードをCに入れてリターンします。親プロセスと子プロセスの関係は常に1対1ですが、適切なプロセスIDを与えることでいくつかのレベルを飛越えて戻ることができます。

子プロセスのファイルハンドルのセットは自動的にクローズされ、親プロセスのファイルハンドルのセットが再びアクティブになります。子プロセスが割り当てていた全てのユーザーRAMセグメントも開放されます。

プロセスIDが0であると部分的にシステムの再初期化が実行されます。全てのファイルハンドルがクローズされ、標準入出力が再オープンされ、全てのユーザーセグメントが開放されます。この後では、コマンドインタプリタは正しい状態ではないためユーザープログラムはコマンドインタプリタに戻ろうとするならこれを行ってはなりません。

このファンクションコールを0F37DHを通して実行されると、B及びCレジスタはエラーコードを返さないことに注意して下さい。これは、プログラムの終了や中断の処理はアプリケーションプログラムによって実行されなければならないからです。エラーコードはアボートルーチンに渡されており、そこにあるプログラムが必要ならばエラーコードを記憶していなければなりません。1次及び2次エラーコードの意味については「エラーコードを伴った終了」ファンクションも参照して下さい。

62H _TERM　エラーコードを伴った終了

　設定　B　終了のエラーコード
戻り値　なし

指定されたエラーコードでプログラムを終了させますが、このコードはエラーなしを示す0であっても構いません。このファンクションはユーザーのアボートルーチンが戻るように設定されていない限り呼び出し側に戻ることはありません。このファンクションの操作は0005Hを通ってMSX-DOSの環境からコールされたのかF37DHを通ってDisk BASICの環境からコールされたのかによって異なります。

0005Hを通ってコールされた場合、ユーザーのアボートルーチンが定義されていると指定のエラーコードと0の2次エラーコードをもってコールされます。このルーチンがリターンするかユーザーのアボートルーチンが定義されていないかすると、制御は0000Hへのジャンプを経由して外部プログラムをロードしたルーチンへ戻されます。これはほとんど常にコマンドインタプリタですが、場合によっては別の外部プログラムの場合もあります。

エラーコードはシステムによって記憶され、次の「親プロセスに戻る」ファンクションがこのエラーコードを返します。

F37DHを通ってDisk BASICの環境からコールされると、制御はBREAKVECTのアボートベクタへ渡されます。この環境では別に定義されたユーザーのアボートルーチンはなく「親プロセスに戻る」はエラーコードを返さないため、エラーコードはBREAKVECTにあるコードによって記憶されなければなりません。

63H _DEFAB アボート終了ルーチンの定義

設定 DE アボート終了ルーチンのアドレス
0000Hで定義解除
戻り値 A 0(エラーを返すことはない)

MSX-DOS の環境で 0005H を通してコールされた時に利用できます。
外部プログラムが何らかの理由で終了しようとしている時には必ずシステムによってアボートルーチンがコールされます。0000H へのジャンプでシステムに戻ろうとした場合はコールされません。MSX-DOS2 のために書かれたプログラムは 0000H へのジャンプではなく「エラーコードを伴った終了」ファンクションで終了しなければなりません。
ユーザーのアボートルーチンは、ユーザーのスタックをアクティブにしファンクションコールが行われた時と同じように IX、IY、と裏レジスタセットをセットし、TPA 全体をページングして起動されます。終了エラーコードは A レジスタ、2 次エラーコードは B レジスタでルーチンに渡され、ルーチンが RET を実行すると A 及び B に渡された値が「親プロセスに戻る」ファンクションで返されるべきエラーコードとしてストアされます。ルーチンはリターンせずに外部プログラム中のなんらかのウォームスタートコードへジャンプしても構いません。システムは完全に安定した状態にあって、どのようなファンクションコールも受け付けることができます。
アボートルーチンが単純にリターンせず、POP HL:RET を実行すると制御はエラーが起こった MSX-DOS コールまたは BIOS コールのすぐ次の命令に渡ります。これは、「ディスクエラー処理ルーチンの定義」と共に使用すると有用で、ディスクエラーが起こった時に現在の MSX-DOS コールをアボートすることができます。

64H _DEFER　ディスクエラー処理ルーチンの定義

　　設定　DE　ディスクエラールーチンのアドレス
　　　　　　　0000Hで定義解除
　戻り値　A　0（エラーは発生しない）

　ディスクエラーが起こった場合にコールされるユーザーのルーチンのアドレスを指定します。TPA全体がページングしてルーチンは起動されますが、ページ3のシステムスタックがアクティブになり、MSX-DOSのファンクションコールが実行された時のレジスタは保存されません。

　エラールーチンはMSX-DOSのコールを実行できますが、エラーの再起を避けるように注意しなければなりません。ファンクション一覧に安全に実行できるファンクションコールが示してあります。標準入出力がリダイレクトされている場合、このルーチンは一時的に無効にした状態でコールされます。

　ルーチン自体のパラメータと結果を下に示します。

　IX、IYおよび裏レジスタセットを含む全てのレジスタは破壊できますが、ページングとスタックは保存しなければなりません。ルーチンはシステムに返らなければならず、外部プログラムにジャンプしてはなりません。外部プログラムに処理を渡すためには、A=1(アボート)を返すことによって、ユーザーのアボートルーチンに制御を渡してアボートルーチン内で必要な処理を行うようにします。

ディスクエラールーチン

　　設定　A　エラーの原因となったエラーコード
　　　　　B　物理ドライブ
　　　　　C　b0　書き込みでセット
　　　　　　　b1　無視の処理が望ましくない時セット
　　　　　　　b2　オートアボートを指示する時セット
　　　　　　　b3　セクタ番号が有効の時セット
　　　　　DE　セクタ番号（Cのb3がセットされている場合）
　戻り値　A　0　システムエラールーチンのコール
　　　　　　　1　アボート
　　　　　　　2　再試行
　　　　　　　3　無視

65H _ERROR　直前のエラーコードの獲得

設定　なし
戻り値　A　0
　　　　B　直前のファンクションコールからのエラーコード

前のMSX-DOSファンクションコールが失敗した原因のエラーコードを見付けることができます。これは、エラーコードを返さない古いCP/M互換のファンクションで使用するためのものです。

66H _EXPLAIN エラーコードの説明

設定	B	説明すべきエラーコード
	DE	64 バイトの文字列バッファへのポインタ
戻り値	A	0
	B	0 あるいは変更無し
	DE	エラーメッセージが入る

MSX-DOS のファンクションから返されるエラーコードについての ASCIIZ 説明文字列を得ることができます。エラーが旧来のファンクションの物である場合は、「直前のエラーコードの獲得」でエラーコードを得なければなりません。

エラーの説明文字列は、システムの言語仕様に基づいたメッセージとなります。

エラーコードが内部の説明文字列を持っている場合は B が 0 にセットされ、説明文字列がないと B は変更されず、返されるメッセージは"システムエラー 194"(40H～FFH)あるいは"ユーザーエラー　45"(00H～3FH)といったものになります。

67H _FORMAT ディスクのフォーマット

設定	B		ドライブ番号(0:カレント、1:A・・・8:H)
	A	00H	選択文字列を返す
		01H～09H	この選択をフォーマットする
		FEH, FFH	ブートセクタの更新
	HL		バッファへのポインタ(A=1～9の場合)
	DE		バッファのサイズ(A=1～9の場合)
戻り値	A	エラー	
	B	選択文字列のスロット(エントリでA=0の場合)	
	HL	選択文字列のアドレス(エントリでA=0の場合)	

DEとHLはディスクドライバによって使用されるバッファ領域を指定します。必要なサイズはわからないので可能な限る大きくするのが最良の方法です。バッファがページ境界をまたぐと、ディスクドライバには一つのページ中の最大の部分を選択して渡されます。

A=FFHの場合、ディスクは実際にはフォーマットされませんが、ディスクに新しいブートセクタを書き込みMSX-DOS2ディスクにします。これは以前のMSX-DOSのディスクをボリュームIDを持つように更新してそれによってMSX-DOS2で可能な完全なディスク検査とファイルの復活を可能にするためです。A=FEHはディスクパラメータのみを正しく更新し、ボリュームIDでブートプログラムを重ね書きすることはありません。

ブートセクタの更新は、「フォーマットの選択の獲得」(A=0)を実行した後でなければならず、これがエラーを返した場合はこれがフォーマットできないドライブで、ディスクはブートセクタの更新で損傷されることがあり、更新を行ってはなりません。

68H _RAMD　RAMディスクの作成あるいは消去

```
設定  B  00H     RAMディスクの消去
         1～FEH  新しいRAMディスクの作成
         FFH     RAMディスクのサイズを返す
戻り値 A  エラー
       B  RAMディスクのサイズ
```

サイズは16KBのセグメントを単位として指定します。Bが1～FEHの場合、既にRAMディスクが存在する場合や未使用セグメントが一つもない場合はエラーが返されます。

69H _BUFFER セクタバッファの割り付け

設定	B	0	バッファ数を返す
		1～FFH	要求するバッファ数
戻り値	A	エラー	
	B	バッファの現在の数	

セクタバッファは最低でも 2 必要です。バッファは通常の 64KB の外のセグメントを使用するため、TPA を減少させることはありません。バッファが多ければ、より多くの FAT とディレクトリセクタを常駐させておけます。バッファの最大数は 20 程度になります。

6AH _ASSIGN 論理ドライブの割り当て

設定	B	論理ドライブ番号(1:A・・・8:H)
	D	物理ドライブ番号(1:A・・・8:H)
戻り値	A	エラー
	D	物理ドライブ番号(1:A・・・8:H)

論理-物理ドライブの割り当てを制御します。

BとDがどちらも0でないと、新しい割り当てがセットアップされます。Bが0でなくDが0であるとBで指定した論理ドライブについての割り当てがキャンセルされます。BとDが両方0であると、全ての割り当てがキャンセルされます。Bが0でなくDがFFHであると、Bで指定した論理ドライブについての現在の割り当てがDに返されます。

ディスクエラールーチンに渡されるドライブ番号は物理番号であるため、対応する論理ドライブと異なる場合があります。

6BH _GENV　環境変数の獲得

設定	HL	ASCIIZ 名前文字列へのポインタ
	DE	値のためのバッファへのポインタ
	B	バッファサイズ
戻り値	A	エラー
	DE	保存される、A=0 の場合バッファが満たされる

指定された環境変数名の現在の値を獲得します。環境変数の大文字、小文字の違いは無視されます。その名前の環境変数が存在しないとヌルが返されます。バッファが小さいと値の文字列は最後のヌルを付けずに切り捨てられ、エラーが返されます。

255 バイトのバッファは常に十分な大きさを持つことになります。

6CH _SENV　環境変数のセット

設定	HL	ASCIIZ名前文字列へのポインタ
	DE	ASCIIZ値文字列へのポインタ
戻り値	A	エラー

新しい環境変数をセットします。環境変数の大文字、小文字の違いは無視されます。

環境変数が有効で、格納するのに十分なメモリがある場合、同じ名前の古い変数が削除され、新しい変数が環境リストの最初に追加されます。値文字列がヌルだと環境変数は削除されます。

6DH _FENV　環境変数の検索

設定	DE	環境変数番号
	HL	名前文字列のバッファへのポインタ
	B	バッファサイズ
戻り値	A	エラー
	HL	保存され、バッファが満たされる

どのような環境変数が現在セットされているかを知るために使用されます。DEの環境変数番号はリスト中のどの変数を検索するかを示します。最初の環境変数が DE=1 に対応します。変数番号<DE>が存在すると、その変数の名前文字列が HL によってさされるバッファ中にコピーされます。バッファが小さすぎる場合には、名前は最後のヌルなしで切り捨てられ、エラーが返されます。255 バイトのバッファは常に十分な大きさを持ちます。変数番号<DE>が存在しないと、ヌル文字列が返されます。

6EH _DSKCHK ディスク検査ステータスの獲得・セット

```
設定  A  00H  ディスク検査ステータスの獲得
         01H  ディスク検査ステータスのセット
      B  00H  有効(A=01Hの場合)
         FFH  無効(A=01Hの場合)
戻り値 A  エラー
      B  現在のディスク検査設定
```

ディスク検査変数はファイルハンドル、FIB あるいは FCB がアクセスされるたびにディスクが変更されたがどうかを見るためにディスクのブートセクタをシステムが再チェックするかどうかを制御します。有効であると、処理の最中にディスクを交換することによって間違ったディスクに誤ってアクセスすることができなくなりますが、有効でない場合にはディスクを破壊する場合もあります。ディスクインターフェースのタイプによって異なりますが、この機能を有効にする場合に若干余分なオーバーヘッドがある場合があります。

しかし、ほとんどのタイプのディスクでは処理時間に変わりはなく、これによって安全性が確保されます。

デフォルトでは有効です。

6FH _DOSVER MSX-DOS のバージョン番号の獲得

設定　なし
戻り値　A　エラー(常に 0)
　　　　BC　MSX-DOS のカーネルバージョン
　　　　DE　MSXDOS2.SYS のバージョン番号

MSX-DOS のバージョンを判断することができます。2 つのバージョン番号が返され、BC に ROM 中の MSX-DOS のカーネルバージョンが、DE に MSXDOS2.SYS システムファイルのバージョンが入ります。これらのバージョン番号のどちらも上位バイトに主バージョン番号、下位バイトに 2 桁のバージョン番号が BCD 値で入っています。例えばバージョン番号 2.34 は 0234H と表現されます。

MSX-DOS1 との互換性のためにこのファンクションを利用するためには次の手順が必要です。ます、エラーが存在する(A<>0)場合はこれは MSX-DOS ではありません。次に B が 2 未満であるとシステムは 2.00 以前のもので C と DE レジスタは不定となります。B が 2 以上の場合 BC と DE はは上記のように指定することができます。

通常、この手順の後、チェックされなければならないバージョン番号は DE の MSXDOS2.SYS のバージョンです。

70H _REDIR　リダイレクションの状態の獲得・セット

設定　A　00H　リダイレクション状態の獲得
　　　　　01H　リダイレクション状態のセット
　　　B　新しい状態
　　　　　b0　標準入力
　　　　　b1　標準出力
戻り値　A　エラー
　　　B　コマンド以前のリダイレクションの状態

主にディスクエラールーチンやリダイレクションに関係なくコンソールに常に出力されなければならないその他のI/Oのために提供されています。

A=1、B=0としてコールすることによって一時的にリダイレクションを無効にすることができます。

このようにリダイレクションを変更すると、システムはいくぶん不安定な状態となり、多くのファンクションコールがこのファンクションを無効にしてリダイレクションをその実際の状態にリセットします。通常、オープン、クローズ、複製等といった、ファイルハンドルを操作する全てのファンクションコールはリダイレクションの状態をリセットします。したがってこのファンクションの効果は完全に一時的なものです。

第 6 章　MSX-DOS2 エラーコード表

　MSX-DOS2 では新設されたファンクションコールからエラーコードが返され、旧来のファンクションでは「直前のエラーコードの獲得」ファンクションコール(65H)でエラーコードを得ることができます。エラーコードが 0 の場合はエラーが発生していない状態を示しますが、それ以外の値の場合はエラーコードがエラーの種類を表しています。

　エラーコードは 0FFH から始まり、値が小さくなっていきます。40H 未満のエラーコードはユーザーエラーで、システムでは使用されず、外部プログラムが独自のエラーを返すために使用できます。また、コマンドインタプリタは 20H 未満のユーザーエラーにはメッセージを出力しません。

　以下にエラーコードとニーモニック、メッセージおよび意味を示します。

６．１　ディスクエラー

　ディスクエラー処理ルーチンに渡されるエラーコードです。またディスクのフォーマットでも返されます。

FFH .NCOMP　このディスクは使用できません　Incompatible disk
　そのドライブではアクセスできないディスクをアクセスした。

FEH .WRERR　書き込み異常です　Write error
　ディスク書き込み中のエラー。

FDH .DISK　ディスクが異常です　Disk error
　原因不明のディスクエラー。

FCH .NRDY　ディスクが入っていません　Not ready
　ディスクドライブが応答しない(通常、ディスクが入っていないことを表す)。

FBH .VERFY　正しく書き込まれませんでした　Verify error
　ベリファイで書き込み後にセクタが正しく読み込めなかった。

FAH .DATA　ディスクのデータが異常です　Data error
　セクタが読めなかった(CRC エラー)。ディスクの損傷を表す。

F9H .RNF　セクターが見つかりません　Sector not found
　セクタがディスク上に見つからなかった。ディスクの損傷を表す。

F8H .WPROT　ディスクが書き込み保護されています　Write protected disk
　書き込み禁止状態のディスクに書き込もうとした。

F7H .UFORM　ディスクがフォーマットされていません　Unformatted disk
　ディスクがフォーマットされていない。

F6H .NDOS　MSX-DOS ディスクではありません　Not a DOS disk
　ディスクのフォーマットが MSX-DOS ではアクセスできない。

F5H .WDISK　ディスクが違います　Wrong disk
　ディスクが別のディスクに交換された。正しいディスクに交換しなければならない。

F4H .WFILE　このファイル用のディスクではありません　Wrong disk for file
　ファイルのオープン中に違うディスクに交換された。正しいディスクに交換する。

F3H .SEEK　シークエラー　Seek error
　ディスクの要求されたトラックが見つからない。

F2H .IFAT　FAT 異常です　Bad file allocation table
　FAT が破壊されている。

F1H .NOUPB
　ディスク交換処理の一部として DOS の内部で処理されるのでユーザーには返らない。

F0H .IFORM　このドライブはフォーマットできません　Cannot format this drive
　フォーマットできないドライブをフォーマットしようとした。

６．２　MSX-DOS ファンクションエラー

通常に MSX-DOS のファンクションコールで返されるエラーです。

DFH .INTER　DOS が異常です　Internal error
起こってはならないエラー。

DEH .NORAM　メモリー不足です　Not enough memory
カーネルデータセグメントのデータを使いつくした。セクタバッファを減らすか環境変数を減らして下さい。
RAM ディスクのための未使用セグメントがない。

DCH .IBDOS　無効な MSX-DOS ファンクション番号です　Invalid MSX-DOS call
不正なファンクション番号でファンクションコールが行われた。
大部分の不正なファンクションコールはエラーを返しませんが、このエラーは「直前のエラーコードの獲得」で返される場合がある。

DBH .IDRV　無効なドライブ名です　Invalid drive
指定のドライブが存在していない。

DAH .IFNM　不正なファイル名です　Invalid filename
ファイル名文字列が不正です。ドライブ・パス・ファイル文字列には返されません。

D9H .IPATH　無効なパス名です　Invalid pathname
ASCIIZ ドライブ・パス・ファイル文字列が渡されるファンクションコールによって返される場合があります。文字列の構文が何らかの形で不正です。

D8H .PLONG　パス名が長過ぎます　Pathname too long
ASCIIZ ドライブ・パス・ファイル文字列が渡されるファンクションコールによって返される場合があります。指定された完全なパスが 63 文字より長いです。使用されている場合はカレントディレクトリも含みます。

D7H .NOFIL　ファイルが見つかりません　File not found
ファイルが見つかりません。ディレクトリが指定されて、それが見つからなかった場合にも返されます。

D6H .NODIR　ディレクトリが見つかりません　Directory not found　ドライブ・パス・ファイル文字列中のディレクトリが見つからなかった。

D5H .DRFUL　ルートディレクトリがいっぱいです　Root directory full
ルートディレクトリ中に空きエントリがないのに新しいエントリが要求されました。

D4H .DKFUL　ディスクがいっぱいです　Disk full
書き込まれようとするデータに対してディスクに十分な領域がありません。

D3H .DUPF　ファイル名が重複しています　Duplicate filename
目的のファイル名が既に目的のディレクトリ中にあります。

D2H .DIRE　ディレクトリが移動できません　Invalid directory move
サブディレクトリをそれ自体の下に移動しようとしました。

D1H .FILRO　ファイルが読み出し専用です　　　　　　　　Read only file
　読み出し専用ファイルに書き込みあるいは削除をしようとしました。

D0H .DIRNE　ディレクトリが空ではありません　　　　　　Directory not empty
　空でないサブディレクトリを削除しようとしました。

CFH .IATTR　無効な属性です　　　　　　　　　　　　　　Invalid attributes
　ファイルの属性を不正な方法で変更しようとしたか、サブディレクトリに対してのみ可能な処理をファイルに行おうとした場合に起こります。また、ボリューム名のFIBの不正な使用によっても起こります。

CEH .DOT　　.や..に対しては操作できません　　　　　　Invalid . or .. operation
　サブディレクトリ中の"."や".."に対して不正な操作をしようとしました。

CDH .SYSX　システムファイルが既にあります　　　　　　System file exists
　既存のシステムファイルと同じ名前のファイルあるいはサブディレクトリを作成しようとしました。システムファイルは自動的には削除されません。

CCH .DIRX　ディレクトリが既にあります　　　　　　　　Directorty exists
　既存のサブディレクトリと同じ名前のサブディレクトリを作成しようとしました。

CBH .FILEX　ファイルが既にあります　　　　　　　　　　File exists
　既存のファイルと同じ名前のサブディレクトリを作成しようとしました。

CAH .FOPEN　ファイルが使用中です　　　　　　　　　　　File is already in use
　そのファイルに対して、既にオープンされているファイルハンドルがあるファイルの削除、名前の変更、移動、あるいは属性や日付や時刻の変更をそのファイルハンドルを使用せずに行おうとしました。

C9H .OV64K　64Kを越える転送はできません　　　　　　　Cannot transfer above 64K
　ディスク転送領域が0FFFFHを越えてしまいます。

C8H .FILE　ファイルの割当異常です　　　　　　　　　　File allocation error
　ファイルのクラスタチェインが破壊されました。

C7H .EOF　　ファイルの終わりです　　　　　　　　　　　End of file
　ファイルポインタが既にエンドオブファイルにある、あるいはそれを越えている場合にさらにファイルから読み込もうとしました。

C6H .ACCV　ファイルアクセス異常です　　　　　　　　　File access violation
　適切なアクセスビットをセットしてオープンされたファイルハンドルに対して読み出しや書き込みを行おうとしました。

C5H .IPROC　無効なプロセスIDです　　　　　　　　　　Invalid process id
　「親プロセスに戻る」で渡されたプロセスIDが不正です。

C4H .NHAND　ファイルハンドルが足りません　　　　　　　No spare file handles
　全てのファイルハンドルがすでに使用中である場合にファイルハンドルをオープンあるいは作成しようとしました。

C3H .IHAND　無効なファイルハンドルです　　　　　　　　Invalid file handle
　指定のファイルハンドルが、システムで許される最大のファイルハンドル番号より大きいです。

C2H .NOPEN ファイルハンドルがオープンされていません File handle not open
指定のファイルハンドルは現在オープンされていません。

C1H .IDEV 無効なデバイスオペレーションです Invalid device operation
デバイスのファイルハンドルやFIBを検索や移動などの不正な操作に使用しようとしました。

C0H .IENV 無効な環境変数です Invalid environment string
環境変数名の文字列に不正な文字があります。

BFH .ELONG 環境変数が長過ぎます Environment string too long
環境変数名あるいはその値の文字列が最大の255文字の長さを越えた、あるいは長すぎてユーザーバッファが足りません。

BEH .IDATE 無効な日付です Invalid date
「日付のセット」に渡されたパラメータが不正です。

BDH .ITIME 無効な時間です Invalid time
「時刻のセット」に渡された時刻が不正です。

BCH .RAMDX RAM DISK(ドライブ H:)は既にあります RAM disk(drive H:) already exists
RAMディスクが既に存在しているのにRAMディスクを作成しようとしました。

BBH .NRAMD RAM DISKがありません RAM disk does not exist
RAMディスクが存在していない時にRAMディスクを削除しようとした。

BAH .HDEAD ファイルが消去されています File handle has been deleted
ファイルハンドルに関連したファイルが削除されたためファイルハンドルはもう使用できません。

B9H .EOL
起こってはならない内部エラー

B8H .ISBFN 無効なサブファンクション番号です Invalid sub-function number
「デバイスI/Oの制御」に渡されたサブファンクション番号が不正です。

B7H .IFCB 無効なFCBです Invalid File Control Block
FCBを使ったファイルアクセスの際、_FOPENなどをコールせずに無効なFCBを使用して読み書きした際に起きるエラーです。

６．３　プログラム終了エラー

　以下のエラーコードはシステムによって内部的に生成され、「中断」ルーチンに渡されるエラーです。通常ファンクションコールからは返されません。中断ルーチンには外部プログラムが「エラーコードを伴った終了」に渡す任意のエラーが渡されることに注意して下さい。

9FH .STOP　Ctrl-STOPが押されました　　　　　　Ctrl-STOP pressed
　”Ctrl-STOP”キーは、全ての文字I/Oなどのほとんどのシステムコールでチェックされます。

9EH .CTRLC　Ctrl-Cが押されました　　　　　　　Ctrl-C pressed
　”Ctrl-C”は、ステータスチェックを実行する文字ファンクションの場合にのみチェックされます。

9DH .ABORT　ディスク入出力が打ち切られました　　Disk operation aborted
　このエラーはディスクエラーがユーザーによってあるいはシステムによって自動的に中断された時に起こります。元のディスクエラーコードは2次エラーコードとしてBレジスタから中断ルーチンに渡されます。

9CH .OUTERR 標準出力でエラーが起きました　　　　Error on standard output
　標準出力チャンネルが文字ファンクションを通してアクセスされている間にエラーが起こった場合に返されます。元のエラーコードは2次エラーコードとしてBレジスタから中断ルーチンに渡されます。このエラーは通常、プログラムが標準ファイルハンドルを変更している場合にのみ起こります。

9BH .INERR　標準入力でエラーが起きました　　　　Error on standard input
　標準入力チャンネルが文字ファンクションを通してアクセスされている間に、エラーが起こった場合に返されます。元のエラーコードは2次エラーコードとしてBレジスタから中断ルーチンに渡されます。このエラーは通常、プログラムが標準ファイルハンドルを変更している場合にのみ起こります。

6．4　コマンドエラー

以下のエラーはDOSのファンクションコールからは返されませんが、コマンドインタプリタによって使用されます。これらは、外部プログラムから利用できますので、ここにのせてあります。

8FH .BADCOM コマンドのバージョンが違います　Wrong version of command
COMMAND2.COMがディスクからその非常駐部分をロードしたが、そのチェックサムが期待通りの値ではありません。

8EH .BADCM　コマンドが違います　Unrecognized command
指定のコマンドが内部コマンドでなく、その名前のCOMおよびBATファイルも見つかりませんでした。

8DH .BUFUL　コマンドが長すぎます　Command too long
バッチファイル中のコマンドが127文字の長さを越えています。

8CH .OKCMD
コマンド行で渡したコマンドが終了した際に返される内部エラーコード。

8BH .IPARM　無効なパラメータです　Invalid parameter
コマンドへのパラメータが何らかの形で不正です。

8AH .INP　パラメータが多すぎます　Too many parameters
コマンドが要求する全てのパラメータを解析した後で、コマンド行上にまだ区切り文字でない文字が残っています。

89H .NOPAR　パラメータが不足しています　Missing parameter
パラメータがあるべきところでエンドオブラインが見つかりました。

88H .IOPT　無効なオプションです　Invalid option
コマンド行で／の後に指定された文字はそのコマンドでは不正です。

87H .BADNO　無効な数値です　Invalid number
数値が求められているところに数字以外の文字が入っています。

86H .NOGELP HELPファイルが見つかりません　File for HELP not found
ヘルプファイルが見つからなかった、あるいはパラメータが有効なHELPパラメータではありませんでした。

85H .BADVER MSX-DOSのバージョンが違います　Wrong version of MSX-DOS
このエラーはコマンドインタプリタで使用されることはありません。コマンドインタプリタでこのエラーが発生した場合は自分自身でもっているメッセージを使用します。しかし、このエラーを返すことが有用な場合には外部プログラムで使用できます。

84H .NOCAT　複写先ファイルは結合できません　Cannot concatenate destination file
CONCATコマンドの目的のファイルがソースの指定に一致しています。

83H .BADEST ファイルが作成できません　　　　　　　　Cannot create destination file

COPYコマンドで目的のファイルが作成されるとソースファイルの一つを重ね書きしてしまいます。

82H .COPY　自分自身にはコピーできません　　　　　　File cannot be copied onto itself

COPYコマンドで目的のファイルが作成されるとソースファイルを重ね書きされてしまいます。

81H .OVDEST ファイルの重ね書きができません　　　　Cannot overwrite previous destination file

COPYコマンドでワイルドカードを使ったソースがワイルドカードを使っておらずディレクトリでなく、デバイスでもない目的ファイルとともに指定されました。

80H～7EHにもメッセージがありますが、マニュアルには書いてないので使わない方がいいでしょう。

参考　CP/M の FCB

参考までに、CP/M での FCB の内容を紹介します。

CP/M では、MSX-DOS でいうところの FAT は、ファイル毎に FCB 内部に記録され、ディスクにもそれが記録されます。このために、ディスクアロケーションマップという領域が FCB 内に存在しています。この部分は MSX-DOS では他の用途に使われています。

CP/M の FCB は 36 バイトですが、MSX-DOS では 37 バイト使用する場合があります。CP/M 互換のファンクションコールでは 36 バイトしか使用しません。

FCB の内容(CP/M)

```
+00,1   ドライブ番号
+01,8   ファイル名
+09,3   ファイル形式
+12,1   ファイル・エクステント番号(通常0)
+13,2   システム用にとってある
+15,1   レコードカウント、あるいは現在のエクステントサイズ
+16,16  ディスクアロケーションマップ
+32,1   カレントレコード
+33,3   ランダムレコード番号(下位バイトが先)_
_
```

第４部 ソフト開発時の注意

第１章　ディスクの環境

１．１　ディスクの有無と数を知る方法

　H．ＰＨＹ DIO(FFA7H)の内容が C9H であればディスクは接続されていません。ディスクから実行されていることが明らかなプログラムではこれを調べることは不必要かつ無意味です。　ディスクが存在する場合は、「ログインベクタの獲得」ファンクションコール(18H)で論理ドライブの数を知ることができます。

１．２　ハードディスクを見分ける方法

ある特定の論理ドライブがハードディスクなのか、それともフロッピーディスクなのかを知るには、次のようにします。
　まず、「ディスクのフォーマット」ファンクションコール(67H)を B にドライブ番号、A に 0(選択文字列を返す)をセットしてコールします。ここで返された、スロットとアドレスからの文字列を得ると、ハードディスクの場合は、文字列が 0 でターミネイトされた直後に、"SCSI7FFC"といった文字列が入っています。最初の 4 文字が ID で次の 4 文字が I/O アドレスを示しています。
　この処理で、「ディスクのフォーマット」ファンクションコールをコールした時、そのドライブがハードディスクインターフェイスだと「このドライブはフォーマットできません」(.IFORM 0F0H)が帰りますが、その後の上記の操作は有効です。
　この方法は DOS1 では使えません。

１．３　物理ドライブ

ソフトウェアから見た時に存在しているドライブを論理ドライブと言いますが、2ドライブシミュレーションによって仮想的に存在しているドライブである場合も含まれています。これにたいして、物理的に存在しているドライブを物理ドライブと言います。

物理ドライブの数や、どのドライブが物理ドライブであるかを知ることは通常できません(と Datapack には書いてあります)。が、MSX マガジンで裏技とでも言うべき方法が紹介されたことがあります。その方法を紹介します。

まず、H.PROMPT(F24FH,3)を書き換えます。ここは通常 RET で、RET すると 2 ドライブシミュレーションのディスク入れ替えメッセージを出力してキー入力待ちをするルーチンがあります。書き換えたジャンプ先ではスタックを 1 レベル捨ててから RET 命令を実行します。これで、2 ドライブシミュレーションのメッセージは出力されません。

次に H:から A:の全てのドライブを逆順にアクセスしていきます。これによって、2 ドライブシミュレーションによって物理ドライブと仮想ドライブが入れ替わっていたりする状態をリセットします。

ここで、A:から H:にむかって今度は順番にアクセスしていきます。このとき H.PROMPT がコールされたらそのドライブは仮想ドライブです。H.PROMPT からのジャンプ先ではコールされたことをワークエリアに記録して RET するようにします。プログラムはディスクアクセスから帰ってきた時にこのワークエリアの内容を調べることによって、そのドライブが仮想ドライブであることを知ることができます。もし、仮想ドライブだった時はもう一度その前のドライブをアクセスして 2 ドライブシミュレーションの状態を戻しておかないといけません。

ここまでのディスクアクセスで、ディスクがドライブにセットされているかどうかはわからないので DISKVE も書き換えて、ディスクエラーを回避しなければなりません。

物理ドライブを調べ終わったら H.PROMPT および DISKVE を元に戻しておいてください。

１．４　マスターカートリッジ

MSX には DISK ROM が最高４つまで接続される可能性があります。ただし、実際に主な処理を受け持つのはそのうち１つだけで、その DISK ROM をマスターカートリッジといいます。そして、マスターカートリッジ以外の DISK ROM をスレーブカートリッジといいます。

　マスターカートリッジのスロット番号は、MASTERS(F348H)に保存されています。

１．５　DISK ROM のスロットを知る方法

FB21H〜FB28H のドライブテーブルという領域に、ディスクインターフェイスに関する情報が記録されています。各ディスクインターフェイスカートリッジ毎に２バイトが割り当てられており、そのカートリッジの論理ドライブ数とスロットアドレスが記録されています。ここで、１番目のカートリッジがマスターカートリッジとは限りません。混同しないようにして下さい。

FB21H	１番目のカートリッジの論理ドライブ数	
FB22H	〃	スロット番号
FB23H	２番目のカートリッジの論理ドライブ数	
FB27H	４番目のカートリッジの論理ドライブ数	
FB28H	〃	スロット番号

１．６　ディスクを使用しない方法

　ディスクを使用できる環境では、ディスクワークエリアの為にフリーエリアが減少してしまいます。そのためにディスクを使用しなくてもよいソフトがディスクワークエリアを確保しない環境で実行を行いたい場合は、システムを起動する際に SHIFT キーを押しながら立ち上げます(SHIFT キー立ち上げ)。このようにすると、DISK ROM は初期化を行わず、ディスクワークエリアは設定されません。当然この状態ではディスクを使用することはできません。

　SHIFT キーの押されていることを確認するとシステムは BEEP 音で知らせます。つまり、

SHIFT キーは BEEP 音がなるまで押していれば良いことになります。

１．７　２ドライブシミュレーションを無効にする方法

　システム立ち上げ時に、BEEP 音がなるまで CTRL キーを押していると 2 ドライブシミュレーション機能を無効にすることができます。システムが立ち上がってから 2 ドライブシミュレーションを無効にすることはできません。

第２章　プログラム作成上の注意

２．１ TPA、フリーエリアの上限

ディスク上で実行されるプログラムの場合、ディスクのワークエリアがメモリ上のどこまでを占めるかはシステムによって異なります。ですからプログラムは実行の前にかならず使用可能なメモリの上限を調べてから実行しなければなりません。もし実行するのにメモリが足りない場合はメッセージを表示して実行を止めて下さい。

DOS 環境の場合、使用できるエリアは TPA の範囲となりますが、TPA の上限は 0006H に示されます。(0006H)からがシステムのある領域なので、(0006H)-1 までを利用できます。

BASIC 環境の場合フリーエリアの上限は CLEAR 文を実行する前の HIMEM(FC4AH)の内容から知ることができます。MSX では DE3FH までのエリアを利用することが推奨されています。ただし、この範囲内でプログラムを書いたとしても、フリーエリアの確認は必ずするようにして下さい。

２．２　環境の確認

プログラムの内容によっては、起動時に動作環境を確認しないと異常動作や暴走につながりますから、次のような項目の確認を怠らないようにしましょう。

MSX のバージョン。MSX1 にない画面モードを使用する場合や、MSX2 以降で追加された機能を使用する場合には MSX バージョンの確認が必要です。MSX のバージョンを確認するには MAIN ROM の 002DH 番地の値を見れば判別できます。この際、この値がいくつならば動作するというプログラムではそれよりバージョンが新しい MSX で使用できなくなってしまいますから、かならずいくつ以上なら動作するというようにプログラムして下さい。

MSX-DOS のバージョン。DOS2 のファンクションコールを使用するプログラムなどでは、DOS のバージョンを判別する必要があります。これも MSX のバージョンの場合と同様、そのバージョン以降ならば動作するというようなプログラムにしておくことが必要です。

画面モード。ANK モードか漢字モードかの判別が必要な場合があります。プログラムの実行に不適当なモードだった場合にはモードを変更する必要があります。また、スクリーンモード、表示色、表示桁数、ファンクションキー表示の有無なども確認する必要がある場合には確認します。

２．３　環境の保存

プログラム実行のためにスクリーンモードなど MSX の動作環境を変更するプログラムでは、DOS のコマンドラインに戻る際にできる限りもとの環境を復元するべきです。

復元するべき環境とは、スクリーンモード、漢字モード、ファンクションキーの表示、表示色、表示文字数、キークリックの有無などです。

2．4　2ドライブシミュレーション

2 ドライブシミュレーションが機能している場合、仮想ドライブにアクセスしようとした時にディスク入れ替えのメッセージが表示されます。しかし、グラフィック画面を使用するプログラムや、画面のフォーマットを乱されては困るプログラムでは、入れ換えメッセージの出力が正常に行われなかったり、画面を乱すなどの問題が発生します。

このメッセージを変える必要のある時は次のようにします。

2 ドライブシミュレーションのメッセージを表示する前に H.PROMPT(F24FH) がコールされます。ここは通常 RET になっていますが、ここにユーザーのディスク交換ルーチンへのジャンプ命令を書きます。

ディスク交換ルーチンが呼ばれた時、A レジスタにドライブ名がキャラクタで入っています。また、スタックトップが標準のプロンプトルーチンへのリターンアドレス、スタックトップ+2 がシステムへのリターンアドレスです。ですから、システムに戻る時はスタックを 1 レベル捨ててからリターンします。

ディスク交換ルーチンではレジスタを保存する必要はありません。またこのルーチンが呼ばれる時にはスロット切り換えは行われないので、ページ 1 に交換ルーチンを置くことはできません。交換ルーチンからユーザープログラムにジャンプしてはいけません。

プログラムが終了又はアボートする時、H.PROMPT の内容を元に戻す必要があります。

2．5　ブレーク処理

　ディスクエラー処理や2ドライブシミュレーションのためにフックを書き換えた場合、プログラムが強制的に中断された場合にもフックを正しく戻せるようにすることが必要です。フックを正しく戻していないと、DOSに戻ってからディスクエラーや2ドライブシミュレーションの処理が発生した時に暴走します。
　ブレーク処理の方法は次のように行います。

DOS1の場合

　WOOM BOOT(0001H,2)を処理ルーチンのアドレスに書き換えます。ブレーク処理ルーチンからシステムに戻ろうとする場合は、WOOM BOOTの内容を元の状態に戻して下さい。さもないと、システムに戻ろうとした時、再びブレーク処理ルーチンに戻ってしまいます。

DOS2の場合

　「アボート終了ルーチンの定義」ファンクションコール(63H)を利用します。システムに戻ろうとする時は、「アボート終了ルーチンの定義」ファンクションコールを利用して、アボート終了ルーチンを解除して下さい。そうしないとシステムに戻るつもりが再びアボート終了ルーチンに帰ってきてしまいます。
　DOS2のアボート終了ルーチンは、0000Hへのジャンプでシステムに戻ろうとした場合にはコールされません。DOS2ではプログラムの終了には「エラーコードを伴った終了」ファンクションコール(62H)を利用しなければなりません。

２．６　エラー発生時の処理

ディスクアクセスで発生するエラーにはロジカルエラーとハードウェアエラーの 2 種類があります。

ロジカルエラーとは、ディスク容量が足りないとか、ファイルがオープンできないといったエラーで、ファンクションコールの戻り値で知ることができます。ユーザーはこの戻り値を見て適当な処理を行うことになります。

これに対して、ハードウェアエラーはディスクが入っていないとか、ライトプロテクトが掛かっているなどといったディスクドライバが返すエラーで、これらのエラーはファンクションコールからエラーコードとして返されることはありません。通常、ディスクドライバからこれらのエラーを受け取ると、DOS のエラーハンドラが"Abort Retry Ignore?"などと問いかけて、その結果により処理を決定します。

問題になるのはハードウェアエラーです。DOS が正常に画面出力できる環境で動作するプログラムであれば、特にプログラムがそれを必要とする場合を除けば、エラー処理は DOS にまかせてもほとんど問題は発生しないのですが、グラフィック画面を使用しているために DOS が文字を出力できない場合や、画面を乱されては困るようなプログラムでは独自にエラー処理を行う必要があります。

ハードウェアエラーの処理方法は DOS1 と DOS2 で異なります。DOS1 のエラー処理を使用したプログラムを、DOS2 で使用して問題が起こることはないと思いますが、DOS2 専用のプログラムや DOS2 での使用をあらかじめ考えて作成されるプログラムでは、DOS2 上で実行された場合は DOS2 でのエラー処理を採用して下さい。

次に DOS1、DOS2 それぞれのエラー処理を説明します。

DOS1 でのエラー処理

DOS のディスクエラーハンドラは、ハードウェアエラーが発生すると、DISKVE(F323H)の示すアドレスの内容が示すアドレスのエラー処理ルーチンを実行します。

エラー処理ルーチンを変更したい場合は、DISKVE の内容を書き換えて、ユーザーのエラー処理ルーチンに処理が渡るようにして下さい。

エラー処理ルーチンのエントリ条件は次のようになっています。

```
 設定  A  エラーを起こしたドライブの番号(0:A 1:B・・・7:H)
       C  エラーの種類
          b0  エラー時の状態
              0:読み込み時  1:書き込み時
          b1～7  エラーの種類
        b7 6 5 4 3 2 1
         1 0 0 0 0 0 0  bad FAT
         0 0 0 0 0 0 0  write protect
         0 0 0 0 0 0 1  not ready
         0 0 0 0 0 1 0  CRC error
         0 0 0 0 0 1 1  seek error
         0 0 0 0 1 0 0  record not found
         0 0 0 0 1 0 1  unsupported media type(読み込み時)
         0 0 0 0 1 0 1  write fault(書き込み時)
         0 0 0 0 1 1 0  other error
戻り値  C  0  ignore(エラーを無視して先へ進む)
          1  retry(入出力を再度試みる)
          2  abort(処理を中断)
```

エラーが発生すると上記のように A にドライブ番号 C にエラーの種類がセットされてエラー処理ルーチンがコールされます。エラー処理ルーチンが、エラーの表示と処理の確認のみ処理して、DOS の標準エラー処理に戻るならば、C に処理を指定してリターンします。

ただし、エラーが bad FAT と unsupported media type だった場合には、ignore を返すとディスクを破壊する場合があるので、必ず abort か retry を返すようにして下さい。

もし、DOS の標準エラー処理を使用せずに独自のエラー処理を行いたいならば、標準エラー処理に帰らずにユーザープログラムにジャンプすることもできます。この場合は、ファンクションコールを実行する前にスタックポインタを保存しておき、エラー処理ルーチンはスタックポインタを元に戻してから処理を続けて下さい。

エラー処理ルーチンの呼び出しの際、Disk BASIC の環境では DISK ROM がイネーブルされていて、スロット切り換えは行われませんのでページ 1 にはエラー処理ルーチンを置くことはできません。DOS の環境ではページ 1 の RAM がイネーブルされてから処理が渡ってきます。DOS1 では、エラー処理ルーチンが DOS のエラー処理ルーチンに戻ろうとする場合は、エラー処理ルーチンの内部でファンクションコールを利用することはできません。

DISKVE を書き換えた状態でユーザープログラムを終了すると、次にディスクエラーが発生した時に正規のエラー処理ルーチンに処理が渡らずに暴走してしまいます。このためにプログラムを終了する際には必ず DISKVE を元の状態に戻すことが必要です。また、Ctrl-STOP でシステムに戻ってしまわないようにしておくことが必要です。これには 0001H と 0002H の内容(WOOM BOOT)を書き換えます。

DOS2 でのエラー処理

DOS2 では「ディスクエラー処理ルーチンの定義」ファンクションコール (64H)があるので、これを利用します。(ファンクションコールの詳細を参照)

この場合、DOS1 の場合とはエラー処理ルーチンの呼ばれた時のレジスタの内容と戻り値の内容がそれぞれ異なりますから注意して下さい。また、エラー処理ルーチンからユーザーのプログラムにジャンプできないことにも注意して下さい。ユーザーのプログラムにジャンプする方法は「ディスクエラー処理ルーチンの定義」ファンクションコールの説明を参照して下さい。DOS2 でのエラー処理ルーチンの内部では、エラー処理ルーチンから安全に呼び出せるとされたファンクションコールを利用することができます。

２．７　コマンドラインでのスイッチ

　DOS のコマンドでスイッチはコマンドの機能を選択、指定するのに使われます。スイッチは"/"で始まりその後にアルファベット(通常 1 文字)を指定します。アルファベットの大文字小文字は通常区別しません。

　スイッチの先頭の文字は"/"を使用するのが原則ですが、実際には"-"(マイナス符号)を使用できるものもあります。"-"を使用できるようにする場合は"/"と"-"の両方を使用できるようにしてください。

　"-"文字はファイル名に使用することができるので、スイッチとファイル名を区別できなくなる可能性があります。ですから、オプションにファイル名を指定するコマンドでは"-"は使用できないようにするなど考慮が必要です。

　また、できるだけ既存のものに準ずることが、使いやすさにつながります。DOS の標準コマンドがよく使用しているスイッチを参考までに以下に挙げます。

/A（アスキーファイル）ファイルを ASCII ファイルとして扱います
/B（バイナリファイル）ファイルをそのまま扱います
/H（不可視属性）不可視属性ファイルも対象にする
/P（ページモード）画面いっぱいになったら出力を一度止めてキー入力を待つ
/V（ベリファイ）コマンド実行中ベリファイ機能を有効にします
/W（ワイド形式）DIR コマンドで複数のファイル名を一行に表示する
/X（無表示）メッセージを表示しない

２．８　バッチの停止

　DOS のバッチコマンドを実行中に、エラーの発生等の原因でバッチ処理を停止したい時は、(0006H)+13H のアドレスに 0 を書き込んで下さい。以後バッチの実行を停止します。

　この方法は DOS2 上では使用できません。

２．９　COMMAND.COM

DOS の外部プログラムを COMMAND.COM という名前にしておけば、DOS の起動時に直接実行させることができます。この場合、スタックがシステムの内部にあるため、スタックを初期化しないと暴走することがあります。スタックを TPA の上限に設定するには LD SP,(0006H) を実行すればいいことになります。

２．10　DOS2 利用上の注意

DOS1 上で動作するソフトはほとんどそのまま DOS2 上でも動作しますが、一部非互換の部分があるために正常動作をしないことがあります。

まず、ファイルアクセスの方式が少し違うために、DOS2 で作成したファイルが DOS1 で読めなくなるようなことがまれにあります。次に、COPY 命令に互換性がありません。COPY 命令や、ディスク-VRAM 間直接転送を使っているソフトは DOS が変わると動作不良を起こすことがあります。これは BASIC でも機械語でも同じです。それから、いくつかのファンクションの機能が若干変更になっているのでこれらのファンクションを使用しているソフトは DOS2 で動かない可能性があります。詳しくは第 3 部第 5 章ファンクションコールの詳細を参照して下さい。

また DOS2 のファンクションでは裏レジスタは保存されますが DOS1 では保存されません。また DOS1 では 0080H からに保存されるパラメータの最後が 0DH でおわりますが、DOS2 では 00H で終わるので注意が必要です。これ以外にも若干、小さなバグのような物がいくらかあるようです。

２．11　スタック

ディスクアクセスの際、スタックはどのページにあっても問題ありません。DOS のファンクションコールは、コールされるとスタックを内部のスタックに切り替えてからファンクションコールを処理します。

第３章　ディスクアクセス

３．１　ディスクアクセスの方法

　ディスクアクセスには、Disk BASIC を使う、システムコールを使う、PHYDIO を使うなどの方法があります。それらの特徴について説明します。

１　Disk BASIC を使う

　特に説明の必要はないと思いますが、Disk BASIC の命令を利用してディスクアクセスを行います。ファイルのアクセスや Ver.2.0 では階層ディレクトリの使用なども簡単に行えるようになっているので、Disk BASIC 上でディスクをアクセスするならばこの方法が最も簡単かつ合理的です。
　この場合、通常はアクセスの途中でディスクエラーが発生すると BASIC のエラーメッセージが表示してプログラムが中断します。ディスクの書き込みを伴うプログラムでは、プログラムが停止した場合やエラーが発生した場合にディスクに正しくデータが書き込まれるような処理を行うと安全です。注意しなければならないのは、ファイルをオープンして書き込みを行い、そのファイルがクローズされないうちにエラーや CTRL-STOP でプログラムが中断した場合です。この時は、まだディスクにバッファを書き出していないので、ファイルのクローズを伴う命令を実行しないうちにディスクを抜いてしまうと、ディスクの内容が破壊される可能性があります。
　ファイルのクローズを伴う命令とは、
CLOSE、CLEAR、CLOAD、END、LOAD、NEW、があります。また、プログラムの書き換えが発生するとファイルはクローズされます。

2　ファンクションコールを使う

この方法はDOS上のプログラムではもっとも一般的です。マシン語を使用すればDisk BASICの環境からでも利用できます。詳しくは第3部を参照して下さい。

この方法では、ファイルのアクセスはもとより、カレントドライブの変更など豊富なDOSの機能が利用できます。

エラーが起こった際の処理等がいくらか複雑になりますが、ディスクの書き込みを伴うソフトでは確実にエラー処理を行って下さい。

3　PHYDIOを使う

PHYDIOとは、MAIN ROMのBIOSエントリに存在する未公開ルーチンですディスクをセクタ単位でアクセスしますが、ディスクドライバに一番近いルーチンなので、メディアIDを指定できたり、ディスクドライバからのエラーコードを得ることができたりということがあり、場合によっては有用です。未公開ルーチンですが、市販ソフトではプロテクトチェックのために一般的に使用されており、とくに問題はないようです。ただしDOS2では使用できません。

ただ、あくまでも未公開ですので、通常のアクセスでは使用しない方が良いでしょう。PHYDIOを使用することによるアクセス速度の改善はほとんどありません。

PHYDIO(0144H/MAIN)(未公開)

```
    設定  A   物理ドライブ番号(0:A・・・7:H)
          B   処理するセクターの数      C   メディアID
          DE  先頭のセクター            HL  転送アドレス
          Cy  0:読み込み  1:書き込み
  戻り値  Cy  0:成功  1:失敗
          A   エラーコード(Cy=1の場合)
          B   残りのセクタ数(Cy=1の場合)
レジスタ  すべて
```

Disk BASICの環境から使用するときは上記の0144Hをコールすればいいのですが、DOSの環境などではPHYDIOのジャンプ先であるFFA7H(H.PHYD)をコールすることも行われます。

３．２　ドライブを止める方法

ディスクアクセスの後のドライブの停止をタイマ割り込みを使って行っているディスクドライバがあります。このようなドライバが接続されているシステムで、割り込みが禁止されたり、システムのタイマ割り込みルーチンを無効にしたりした場合に、ディスクドライブが止まらなくなる現象が発生します。

これを防ぐためには、ディスクアクセスの後にマスターカートリッジの 4029H をコールして下さい。ただし、古い DISK ROM にはこのルーチンがない場合もありますので、コールする前にその存在を確認する必要があります。4029H の内容が 00H だった場合はモーターを停止するルーチンは存在しません。よって、モーターを停止することはできません。

３．３　フォーマット

ディスクのフォーマットをする方法を説明します。DOS2 の場合はシステムコールでサポートされていますので第 3 部第 5 章を参照して下さい。

```
4025H/マスタースロット
        設定  HL  フォーマット用ワークエリアの先頭アドレス
              DE  フォーマット用ワークエリアの長さ
      戻り値  Cy 0:成功  1:失敗
              A  エラーコード(Cy=1の場合)
```

このルーチンをコールすると BASIC で_FORMAT を実行したのと同じようにフォーマットのタイプを聞いてきます。ここでフォーマットのタイプを入力すると実際にフォーマットします。フォーマットの指定を入力することなしにフォーマットすることはできません。

フォーマット用ワークエリアは 8KB 以上あれば問題ありません。

FORMAT(0147H/MAIN)やフックの FFACH(H.FORM)をコールしても同様にフォーマットすることができますが、こちらは未公開です。

３．４　turboR でのディスクアクセス

turboR で、DOS1 または Disk BASICVer.1.0 上で動いているプログラムが高速モードになっている場合にディスクアクセスをする際には CPU を Z80 に切り換えてからディスクアクセスをしなければなりません。なぜかというと、ディスクドライバは Z80 上で動作することを考えて作られているので、R800 を使っている時にディスクアクセスを行うと FDC とのタイミングが狂う可能性があるからです。DOS2 上では、DOS カーネルがディスクアクセスを行う時には Z80 に切り換えているので問題は発生しません。

３．５ 効率のよいディスクアクセス

ディスクアクセスのアルゴリズムを間違えると、必要以上にディスクアクセスに時間が掛かってしまい、使いにくいソフトになってしまいます。実用的なソフトでは、ファイルアクセスの方法もデータの構造や目的によりある程度制限があると思いますが、ゲーム等のソフトでは、ディスクアクセス待ちは短いほうが良いので、ディスクアクセスの方法を工夫すると良いでしょう。ディスクアクセスの効率を上げるためには次のような事を守って下さい。

まず、データの読み込みや書き込みはなるべく大きなまとまりを単位としてすることです。例えば、1 セクタ(512 バイト)づつ 16 回読み込みを行うよりも、1 回で 16 セクタ(8KB)の読み込みを行ったほうが一般に早くなります。これは、ディスクの構造を考えればわかるのですが、1 セクタを読む度にユーザーに処理を返していると、ユーザーが処理を行っている間にディスクは回転して次のセクタはヘッドの位置を通過してしまい、ディスクがもう一周してくるのを待つことになってしまいます。ディスクドライバの内部で処理しているぶんには、次のセクタがヘッドの位置にくるまでに次のセクタを読み込む処理に移れるので無駄がありません。この読み込み処理の時間はディスクドライバの性質によって大きく変わることがありますが、セクタ単位でアクセスすると最悪の場合で最高の場合の数倍程度の時間が掛かることがあります。

第４章　プログラムのオートスタート

　ゲームソフト等で、自作のプログラムをオートスタートするようにしたいと考えることがあると思います。そのような場合に使える代表的な方法を説明します。

４．１　BASIC の場合

　BASIC のプログラムをオートスタートさせたい場合は実行させたい BASIC プログラムを"AUTOEXEC.BAS"というファイル名でセーブしておくことです。こうしておけば、BASIC が起動した直後にそのプログラムが読み込まれて実行されます。

４．２　DOS プログラム

　DOS の外部コマンドを自動的に実行させたい場合は"AUTOEXEC.BAT"という名前のバッチファイルで実行させたい外部コマンドを実行させます。
　コマンドインタプリタに戻る必要がない場合は、外部コマンドを"COMMAND.COM"という名前にして置くことで MSXDOS.SYS によって読み込まれて実行されます。この場合スタックの初期化を忘れずに行うことが必要です。

４．３　ブートセクタからの起動

以上の方法を使うと、実行の前に BASIC や DOS の起動メッセージが出てしまいますが、これを避けたい場合にはディスクのブートセクタに起動プログラムを書いておくことにより DOS などの起動メッセージなしに自分のプログラムを自動実行させることができます。

ブートセクタから起動する場合にはブートセクタの C01EH 番地からにプログラムを書いておきます。ブートセクタの先頭 256 バイトはシステムの起動時に C000H に読み込まれ、2 回コールされます。

1 回目のコールでは Cy フラグをリセットしてコールされ、その時のスロットの状態はページ 0 が BIOS ROM、ページ 1 が DISK ROM、ページ 2 と 3 は RAM になっています。この時 A が 0 ならシステムが起動した直後であることを表します。また、HL には DISKVE のアドレスが入っています。

2 回目のコールでは Cy フラグがセットされてコールされ、その時のスロットの状態はページ 1 が DISK ROM で他のページは RAM になっています。この時も 1 回目のコールと同様にレジスタがセットされ、A が 0 なら起動直後を表し、HL に DISKVE のアドレス、DE にページ 1 を RAM に切り換えるルーチンのアドレスがセットされています。

1 回目のコールではページ 1 を RAM に切り換えることはできないのでユーザープログラムの起動には適当ではありません。
CO1EH からのプログラムではまず"RET NC"で 1 回目のコールを無視した後、2 回目のコールで実行するプログラム本体を読み込んで実行します。

ブートセクタの読み込みでは先頭の 256 バイトのみが読み込まれるのでブートプログラムはこの範囲で書く必要があります。

第5部 ディスクシステムの内部

第１章　ディスク

１．１　フロッピーディスク

フロッピーディスクは、磁性体を塗った円盤状のフィルムにデータを記録するもので、ディスクという名前の付いている物としては最も一般的な物です。様々なフォーマットの物がありますが、MSX では現在 3.5 インチ 2DD フォーマットの物が主流になっています。

2DD では、メディア ID は F9H、1 セクタ 512 バイトで 1 クラスタが 2 セクタつまり 1024 バイトです。総セクタ数は 1440 です。

2DD ディスクで"2"はディスクの両面を使用していることを示します。それぞれの面をサイドといって、番号を付けてサイド 0、サイド 1 といいます。1 つのサイドには、同心円状のトラックがディスクの外側から内側に向かって 80 本あります。トラックを表裏で一組にしてシリンダと呼びます。片面ディスクでは、1 シリンダ=1 トラック、両面ディスクでは 1 シリンダ=2 トラックになります。シリンダは外側から内側に向かって 0〜79 の番号がふってあります。トラックは両面ディスクでは 80*2 で 160 トラックあり、トラック番号は、(シリンダ番号)*(面の数)+(サイド番号)で表され、トラック 0〜159 があります。それぞれのトラックには 9 セクタフォーマットの場合は 9 つのセクタがあり、ディスク全体では、9(セクタ)*160(トラック)=1440(セクタ)となります。

１．２　RAM ディスク

MSX-DOS2 で、RAM ディスクができました。RAM ディスクは DOS 上では RAMDISK コマンドによって作成され、物理ドライブ名は常に H:となります。アクセスが非常に高速なので、フロッピーディスクから RAM ディスクにプログラム等をコピーすると快適な操作ができます。

RAM ディスクはマッパ RAM に取られます。ですから、マッパ RAM に空きがないと RAM ディスクを確保することはできません。

RAM ディスクのフォーマットは、1 セクタが 512 バイトです。容量が 2MB までの時、1 クラスタが 1 セクタとなっています。ですから、1 クラスタは 512 バイトとなります。容量が 2MB 以上になると 1 クラスタは 1024 バイトとなります。

RAM ディスクのメディア ID は FFH となっています。

１．３　ハードディスク

ハードディスクは、フロッピーディスクと同じように、磁性体を塗った円盤にデータを記録します。フロッピーよりもアクセスが高速で、容量も大きくなっています。

MSX では、アスキーから発売されている HD インターフェイスを利用することによってハードディスクを使用できます。

ハードディスクのクラスタは領域の大きさによって 1KB から 64KB まで変化します。

第２章　DOS カーネル

２．１　DOS カーネル

１　DOS カーネルの役割

DOS カーネルとは、DOS の中核をなすプログラムのことで、DISK ROM の内部に存在しています。DISK ROM の内部には、DOS カーネル、Disk BASIC、ディスクドライバが一緒に入っています。

DOS カーネルは、ディスクアクセスの全てに渡って、ユーザープログラムとディスクドライバの橋渡しをしており、統一的なディスク操作環境をユーザーに提供します。ユーザーはディスクアクセスの際に直接 I/O ポートやディスクドライバにアクセスすることはなく、かならず DOS のファンクションコールを利用してディスクにアクセスします。

２　DOS カーネルのバージョン

DOS カーネルには大きく分けて、DOS1 カーネルと DOS2 カーネルの 2 種類があります。

DOS2 カーネルは基本的には DOS1 の上位互換なので利用方法等はどちらも変わりませんが、内部的には大きく異なっています。

また DOS1、DOS2 のそれぞれのカーネルの中でも幾つかのバージョンが存在しています。バグの修正や、機能追加によってそれぞれ少しづつ変わっています。DOS2 では「MSX-DOS のバージョン番号の獲得」ファンクションコール(6FH)でカーネルのバージョンを知ることができます。DOS1 ではバージョンを知ることはできません。

２．２　DOS1 カーネル

１　メモリマップ

ディスクのない状態では F380H からがシステムワークエリアで F37FH までをユーザーが使うことができました。ディスクが接続されると、まずディスクシステムのために F1C9H までがワークエリアとして確保されます。つづいて、各 DISK ROM が自分で使うワークエリアを確保します。ここで確保されるワークエリアの大きさは、ディスクドライバによって異なるのでメーカーが違うとアドレスも変わってきます。DPB が DISKROM ごとに 15H*ドライブ数分確保されます。ディスクドライバ用ワークエリアと DPB がすべての DISK ROM について確保されます。次に DOS のセクタバッファが 3*全てのディスクドライブで最も大きいセクタサイズのワークエリアが確保されます。続いて FAT 用のエリアが確保されます。それぞれの FAT はそのドライブのデフォルトの DPB をもとに、セクタサイズ*FAT のセクタ数+1 バイト確保されます。先頭の 1 バイトはメモリ上の FAT とディスク上の FAT が一致しているかのフラグです。まだディスクから読み出されていない時 FFH、メモリ上の FAT を変更してまだディスクに書き込んでいない時 1、メモリ上の FAT とディスク上の FAT が一致している時 0 になります。

DOS の場合、ディスク用のワークエリアはここまでですが、Disk BASIC の場合さらに Disk BASIC が使う FCB と、BLOAD、BSAVE のためのルーチンのエリアが確保されます。

BASIC から DOS に入った場合でも、このエリアは開放されません。

DOS1 で、通常ディスクインターフェース 1 つで 2 ドライブ(シミュレーション)のシステムの場合、ディスクのワークエリアの先頭(HIMSAV)は DF9DH-(そのインターフェースのディスクドライバのワークエリアサイズ)となります。

ディスクを接続した環境でのメモリマップ

BASICが使用するエリア	(BLDCHK+1/F378H) (FCBBASE/F353H)	BLOAD,BSAVEルーチン　19Hバイト FCBエントリ　25H*(FILMAX)バイト
FATエリア	(HIMSAV/F349H)	ドライブB　FAT管理バイト ドライブB　FAT ドライブA　FAT管理バイト ドライブA　FAT
バッファ	(DIRBUF/F351H) (BUFFER/F34FH) (SECBUF/F34DH)	DOSディレクトリ用セクタバッファ DOS汎用セクタバッファ ドライバ用セクタバッファ
1番目のカートリッジのワーク	(DPBLIST/F357H) (DPBLIST/F355H)	ドライブA　DPB ドライブB　DPB ディスクドライバ用ワークエリア
ディスク用ワークエリア	F1C9H	
システムワークエリア	F380H	

2　ドライブワークエリア

ディスクドライブが接続されている状態ではシステムワークエリアのほかにF1C9H〜F37FHにディスク用のワークエリアが確保されます。その内容はほぼ未公開ですが解析によってわかった部分を紹介します。この内容はDOS2では変更になっている場合があります。

後半には、システムワークエリア内でDOSがアクセスするワークエリアを紹介します。

初期化時にF1C9H〜F37FHを00Hで埋めた後、F1C9H〜F236Hに7397H〜7404Hのデータが転送されます。

```
F1C9H　サブルーチン　文字列出力
F1D9H　サブルーチン
F1E2H　サブルーチン
F1E8H　サブルーチン
F1F4H
F1F7H　システム定義デバイス名
F1FFH　”NUL ”
F216H,1
F221H,2
F223H,2
F22CH,1　閏年のチェックのフラグ　1CH:閏年ではない　1DH:閏年
F237H,1　CTRLキーチェック
F23BH,1　プリンタ出力のフラグ　0:無　1:有
F23CH,1
F23DH,2　DTAアドレス　システムコール1AHで設定される
F23FH,2
F241H,1
F243H,2
F245H,1
F246H,1
F247H,1　デフォルトドライブ番号
F248H,3　日付の保存　（日、月、年の順）
F24CH,2
F24EH,1
```

○フックエリア(アドレスの後にコール元アドレスを表示)

```
F24FH,3 625FH H.PROMPT　2ドライブシミュレーションメッセージ表示
F252H,3 41F4H
F255H,3 425AH
F258H,3 42BCH
F25BH,3 4317H
F25EH,3 4348H
F261H,3 440EH ドライブ番号処理
F264H,3 4471H
```

F267H,3 44DBH
F26AH,3 4553H/4555H DPB アドレスを得る
F26DH,3 45CFH
F270H,3 46C5H ディスクリード
F273H,3 470AH
F276H,3 4748H
F279H,3 4755H ディスクライト
F27CH,3 4916H HLBC=DE*BC
F27FH,3 4932H
F282H,3 4989H
F285H,3 49B1H
F288H,3 4A36H
F28BH,3 4A46H
F28EH,3 4B56H
F291H,3 4BE2H
F294H,3 4C22H
F297H,3 4C97H
F29AH,3 4D05H
F29DH,3 4D8CH
F2A0H,3 4E48H
F2A3H,3 4EDBH
F2A6H,3 4F12H
F2A9H,3 4F9EH
F2ACH,3 5104H
F2AFH,3 53A8H CTRL コードのチェック
F2B2H,3 5496H
F2B5H,3 5523H CLOCK の年の処理

F2B8H,1
F2C4H,1
F2DCH,1
F2DDH,1
F2DEH,1
F2DFH,1
F2E0H,1
F2E1H,1 　ドライブ番号
F2E2H,2
F2E4H,2
F2E6H,2
F2E8H,2
F2EAH,2
F2ECH,2
F2EEH,2
F2EEH,1
F2FFH,1 　リード／ライトのフラグ？
F300H,2
F304H,2 　SP の保存
F306H,1 　システムコール時のリターンパラメータに関するフラグ
　　　　　　0 でなければ LD HL,BA の後 RET
F307H,2 　次のエントリの検索のための FCB のアドレス保存
F30CH,1
F30DH,1 　RAWFLG　0 以外ならベリファイを行う
F30EH,1 　日付表示のフォーマットタイプ
F304H,4 　MAINROM の 0034H からの値

```
F323H,2  DISKVE  ディスクエラー処理ルーチンへのポインタのポインタ
F325H,2  BREAKV  CTRL-C 処理ルーチンへのポインタのポインタ
F327H,5  F372H フック     LD A,1AH:RET
F32CH,5  F375H フック     RET
F331H,5  F37DH フック BASIC のファンクションコール 56D3H/マスターカートリッジへ
F336H,1  入力されたキーがあれば 0FFH、F337H にキャラクタ
F337H,1  入力されたキーのバッファ
              CTRL-STOP が押されると F336H と F337H に 03H
F338H,1  CLOCK-IC の有無  0H:無し  FFH:有り
F339H,2  フォーマットでのスタックポインタの保存
F33BH,2
F33FH,1  システム立ちあげ時には CTRL キーが押されていれば 2
         ドライブ番号の保存
F340H,1  0 ならパワー ON 直後
F341H,1  RAMAD0  ページ 0 の RAM のスロット番号
F342H,1  RAMAD1  ページ 1 の RAM のスロット番号
F343H,1  RAMAD2  ページ 2 の RAM のスロット番号
F344H,1  RAMAD3  ページ 3 の RAM のスロット番号
F345H,1
F346H,1
F347H,1  論理ドライブの数
F348H,1  MASTERS  マスターカートリッジのスロット番号
F349H,2  HIMSAV  DOS カーネルの動作に必要なエリアの底のポインタ
F34BH,2
F34DH,2  SECBUF  ドライバ用セクタバッファのアドレス
F34FH,2  BUFFER  DOS 汎用セクタバッファのアドレス
F351H,2  DIRBUF  DOS ディレクトリ用セクタバッファのアドレス
F353H,2  FCBBASE Disk BASIC の FCB エリアのアドレス
F355H,16 DPBLIST  各ドライブの DPB のポインタ
```

○ジャンプテーブル

```
F365H,3  基本スロット選択レジスタを読む  IN A,(0A8H)
F368H,3  ページ 1 が RAM になる
F368H,3  ページ 1 切り換え
F36BH,3  F1C9H/F1D9H/F1E2H/F1F0H
F36EH,3
F371H,3  JP 0F327H
F374H,3  JP 0F32CH
F377H,3  BLDCHK  Disk BASIC の BLOAD、BSAVE ルーチンへのジャンプ
F37AH,3
F37DH,3  Disk BASIC のファンクションコールエントリ  JP 0F331H
```

〇システムワークエリア

F4D9H,1
F55BH,1
F568H,17
F674H,2　STKTOP　BASICがスタックとして使用する番地
F6ABH,2　セクタバッファの最大サイズ。初期化の時に使用される
FB21H,8　各ディスクインターフェイスの管理する論理ドライブの数とそのディスクインターフェイスのスロット番号。ただし空いたテーブルには0が入ります

1番目のカートリッジ	論理ドライブの数
:	スロット番号
:	:
4番目のカートリッジ	論理ドライブの数
	スロット番号

FB29H,12 各ディスクインターフェイスカートリッジ毎の割り込みアドレス保存

1番目のカートリッジが保存する割り込みルーチン	スロット番号
:	アドレス
:	:
4番目のカートリッジが保存する割り込みルーチン	スロット番号
	アドレス

FC48H,2　BOTTOM 実装したRAMの最低番地MSX2以降8000H
FC4AH,2　HIMEM 利用可能なメモリの上位番地
FD99H,1　DISKID
　　初期化の際に使用される
　FFH　RAMが16KB未満/SHIFT立ち上げ

〇フックエリア(マスタースロット内のジャンプ先を表示)

FDEFH,5　6B96H H.DSKO
FDF9H,5　6F20H H.NAME
FDFEH,5　6F00H H.KILL
FE08H,5　707BH H.COPY
FE12H,5　7061H H.DSKF
FE17H,5　6B75H H.DSKI
FE21H,5　6CD7H H.LSET
FE26H,5　6CD6H H.RSET
FE2BH,5　6C49H H.FIEL
FE30H,5　6DAFH H.MKI$
FE35H,5　6DB2H H.MKS$
FE3AH,5　6DB5H H.MKD$
FE3FH,5　6DD7H H.CVI
FE44H,5　6DDAH H.CVS
FE49H,5　6DDDH H.CVD
FE4EH,5　66A4H H.GETP
FE58H,5　66B3H H.NOFO
FE5DH,5　66FCH H.NULO
FE62H,5　68D0H H.NTFL
FE71H,5　690EH H.BINS
FE76H,5　6939H H.BINL
FE7BH,5　6E88H H.FILE
FE80H,5　6BDAH H.DGET
FE85H,5　688EH H.FILO
FE8AH,5　6819H H.INDS
FE99H,5　700DH H.LOC
FE9EH,5　7009H H.LOF
FEA3H,5　6E70H H.EOF

```
FEADH,5  6875H H.BAKU
FEB2H,5  7323H H.PARD
FEB7H,5  737CH H.NODE
FEBCH,5  H.POSD INC SP:INC SP:JP 6F1DH
FED5H,5  H.LOPD
FEFDH,5  71D3H H.ERRP
FFA7H,5  6055H H.PHYD PHYDIO
FFACH,5  606BH H.FORM FORMAT
```

3　内部解析資料

DOS1 カーネル解析結果を紹介します。

●ディスクドライバ　ジャンプテーブル
4010H　DSKIO
4013H　DSKCHG
4016H　GETDPB
4019H　CHOICE
401CH　DSKFMT
401FH　MOTOR_OFF_ENTRY

●DOS カーネル　ジャンプテーブル
4022H　BASIC コールドスタート（JP 5B3AH）
4025H　フォーマット（SCF:JP 60B1H）
4029H　モーターストップ（JP 620DH）
402CH　空き　00H
402DH　GETSLT　スロット番号を得る（JP 5FB3H）

●基本ルーチン群
4030H　HL に F34BH の内容をセット
4034H　Ctrl-STOP キー判定とキー入力
4078H　キー入力
408FH　CHPUT(00A2H/MAIN)をコール
409BH　CHGET(00A5H/MAIN)をコール
40A7H　ファンクションコール 00H／BASIC ウォームスタート
　　　　MAIN-ROM の 409BH をコール
40ABH　MAIN-ROM の(IX)をインタースロットコール
　　　　戻り値 IX　コールするアドレス
40B8H　CLOCK-IC の存在を調べる
　　　　戻り値　あれば(F338H)に FFH
　　　　　　　　24 時計、アラーム OFF にセットされる
4115H　CLOCK-IC に年月日を書き込む
4130H　CLOCK-IC の秒以前のリセット
4142H　CLOCK-IC のレジスタ E からに D、C、B の順に書き込む
414EH　CLOCK-IC のカウントを開始する
4159H　CLOCK-IC のブロック 0 を選択しカウントを停止する
4160H　CLOCK-IC のレジスタ E からに H の値を BCD に変換して書き込む
4179H　CLOCK-IC から日付と時間を読み込む
41ADH　CLOCK-IC のレジスタ(E-1)(E-2)を 2 進数にして A に入れる
41C1H　MSX-DOS Ver2.2 の文字列

●ファンクションコール処理ルーチン
41EFH　ファンクションコール 0CH　バージョン番号を得る
　　　　戻り値 BA 0022H
436CH　ファンクションコール 13H
4392H　ファンクションコール 17H
4427H　デフォルトドライブ番号処理
4462H　ファンクションコール 0FH
4555H　DPB のアドレスを得る
4916H　HLBC=DE*BC

576FH　ディスクシステムのINITルーチン
5897H　H.RUNCからジャンプしてくるルーチン
5B1BH　"AUTOEXECBAS"
5B27H　'RUN"AUTOEXEC.BAS"'の文字列
5B3AH　BASICコールドスタート
5C6EH　フックをセット
5C96H　フックのデータ　～5D23H
5EC8H　ワークエリアを確保する
　　　　メモリが不足したときは、メッセージを表示して停止
5ECCH　メモリ不足の表示をして停止
5EE8H　ワークエリアを確保
　　　　設定 HL　バイト数
5F86H　インラインパラメータで文字列を表示
5F8CH　文字列を表示
　　　　設定 HL　文字列の先頭番地（文字列は00Hで終わる）
5FB3H　スロット番号を得る
　　　　戻り値 A　スロット番号
5FC2H　GETWRK SLTWRK(FD09H～)の内容を得る
　　　　戻り値 HL=IX　SLTWRKの値
5FCDH　SLTWRKのそのスロットに対応する番地を得る
　　　　戻り値　HL　SLTWRKの番地
5FE7H　EXPTBLのそのスロットに対応する番地を得る
　　　　戻り値　HL　EXPTBLの番地
5FF1H　BC=A HL=HL+A
5FF6H　各インターフェイスのタイマ割り込み処理をセットする
　　　　設定　HL　タイマ割り込みルーチンの番地
6027H　ディスクドライバの割り込み処理後に保存してあったフックへ処理を渡す
6055H　PHYDIO
6067H
6086H　論理ドライブ番号からそのドライブを管理するインターフェイスのスロットと
　　　　インターフェイス内のドライブ番号を得る
　　　　入力　　A　ドライブ番号
　　　　出力　　IYH　スロット番号
　　　　　　　　A　インターフェイス内のドライブ番号
609AH　ドライブ(A 0..7)のメモリ上のFATを無効にする
60B0H　FORMAT(BIOS)
60B1H　フォーマット(4025H)
6166H　キーバッファをクリアして1文字入力
6174H　メッセージ表示してキー入力待ち、アボートあり
62C6H　文字を出力して改行
62C9H　改行

第３章　ディスクドライバ

MSX ではディスクドライブのインターフェイスは統一されていないため、ディスクドライバの内容は各機種によって異なっています。そのため、ディスクドライブに対して直接操作をすることは禁止され、またたとえしようと思っても I/O ポートなどの情報は公開されていません。ここでは、独自の調査によりそれらの情報を紹介します。

３．１　ディスクドライバ

１　ディスクドライバ

ディスクドライバはその名の通りディスクドライブを駆動するためのソフトウェアで、DOS と FDC 及びディスクドライブの間の橋渡しをしています。普通、ディスクをアクセスするときは MSX-DOS のシステムコールを利用するわけですが、MSX-DOS は最終的なディスクアクセスの段階でディスクドライバに処理を渡します。ディスクドライバは DOS の下で最も末端の処理を行うものです。DOS カーネルのレベルでは FDC やドライブに対して直接アクセスすることはありません。

ディスクドライバと呼ばれるプログラムには、ドライブの初期化、セクタ単位でのディスク入出力、フォーマット、DPB データの提供などの機能があります。

ディスクドライバは DISK ROM 内にあり、DOS カーネルと一緒に DISK ROM に書き込まれています。ディスクインターフェイスが 2 つ以上ある場合、すなわち DISK ROM が 2 つ以上ある場合には当然ディスクドライバも複数あり、それぞれが自分の属するインターフェイスに接続されているドライブを管理します。

ディスクドライバは DISK ROM の 7405H～7FFFH にあります。ドライバが提供するルーチンは仕様で決められています。4010H～4021H にディスクドライバにジャンプするジャンプテーブルがあり、DOS はここをコールしています。というのは、ドライバが複数ある場合のために DOS は論理ドライブ番号をもとにそれ

ぞれのドライバを呼び出すようになっているためです。そのほか、ディスクドライバ内にはドライブの初期化などのルーチンがあります。しかし、こちらはジャンプテーブルを介すことなく、DISK ROM 内の初期化ルーチンから直接呼び出されます。初期化はそれぞれの DISK ROM によって行われるからです。ディスクドライバの割り込みルーチンもジャンプテーブルを介さず呼び出されますが、これは初期化の際にワークエリアに記録されたアドレスを利用しています。

ディスクドライバのプログラムは当然メーカーまたは機種によって内容が異なっているわけですが、基本的な部分はメーカーが異なっていても共通点が多数あります。

それでも末端の I/O 関係ではメーカーが異なればハードウェアが異なるためかなり変わらざるを得ません。例外としては OEM 生産のために別のメーカーと内容が同じである場合や、経緯はわかりませんが明らかに同じと思われるプログラムを使っている場合があります。

2　ディスクドライバ用ワークエリア

インターフェイスが管理するドライブの数やドライブの状態などを保存するために、ディスクドライバ毎にワークエリアが確保されています。ドライバが必要とするワークエリアのサイズと内容はドライバによって異なりますから、ワークエリアは仕様によって固定されているのではなく、各ドライバがフリーエリアから確保しています。

よってそのアドレスはシステムの構成によって変化することがあります。

ディスクドライバ用ワークエリアはシステムの初期化の際に確保され、以後固定されます。ワークエリアは各ドライバごとに確保され、その先頭アドレスは SLTWRK(FD09H～)のその DISK ROM の存在するスロットに対応する場所に記録されています。

この先頭アドレスを得るためには多少なりとも複雑な処理が必要となりますが DISK ROM が表に出ている状況では、内部ルーチン GETWRK(5FC2H)が利用できます(DOS2 上では不可)。

ワークエリアの内容はドライバによって異なります。ですからこのワークエリアを直接参照することは、使用機種を特定しないかぎりできません。

3　ディスクドライバの提供するルーチン

ディスクドライバの提供するルーチンについて解説します。まずは、4010H～4021H のジャンプテーブルから呼び出されるルーチン、続いて初期化ルーチンから直接呼び出されるルーチンを解説します。

このジャンプテーブルから呼び出されるルーチンは BIOS と同様に仕様が取り決められているので、直接コールしても平気なはずです(ただし未公開、保証はない)。ただし、ディスクドライバを直接コールするためには論理ドライブ番号から、そのドライブを管理するディスクドライバのスロットとインターフェイス毎の論理ドライブ番号を割り出す必要があり、面倒な手順が必要になります。独自の DOS を作成する場合など、ディスクドライバを直接コールすることによって、論理的なフォーマットレベルではオリジナルフォーマットを使用することも可能な訳です。しかし可能な限りは使わない方が良いでしょう。

DSKIO(4010H)

設定 A ドライブ番号(0～)
Cy 0ならリード、1ならライト
B リード／ライトするセクタの数(1～255)
C メディアID
DE 先頭セクタの論理セクタ番号(0～)
HL 転送アドレス
戻り値 エラーならCy=1さらにAにエラーコードBに読み残したセクタ数
エラーコード
0 Write protected
2 Not ready
4 Data (CRC) error
6 Seek error
8 Record not found
10 Write fault
12 Other errors
レジスタ AF,BC,DE,HL,IX,IY

ディスクをセクタ単位で読み書きします。ディスクドライバ内ルーチンであるため、ドライブ番号は0～1である必要があります。当然、あるドライブにアクセスしようとする時はそのドライブを管理するディスクドライバのスロットのルーチンを呼ばなければなりません。2ドライブシミュレーション機能はディスクドライバが管理するため、このルーチンにおいてもその機能は有効です。

DSKCHG(4013H)

```
  設定  A  ドライブ番号
        B  0
        C  メディアID
        HL DPBのアドレス
戻り値  成功  Cy=0
              B  ステータス
                 1        ディスクが入れ替えられた
                 0        不明
                 -1(FFH)  入れ替えられていない
        失敗  Cy=1
              A  エラーコード(DSKIOと同じ)
レジスタ  AF,BC,DE,HL,IX,IY
```

ディスクの入れ替えが発生したかどうかを調べて返します。インターフェイスがディスクの入れ替えを判別できないシステムではディスクの入れ替えを、ディスクアクセス後に一定時間をカウントすることで行うものもあります。

システムはディスクの入れ替えが発生すると、FATの先頭バイトを読み、それによってDPBをセットし直します。

GETDPB(4016H)

設定　A　ドライブ番号
B　メディアID(FATの先頭の値)
C　メディアID
HL DPBの先頭アドレス
戻り値　DPBの1バイト目および最後の2バイトを除いた18バイトにデータをセット
レジスタ　AF,BC,DE,HL,IX,IY

メディアIDから実際のDPBをセットします。

CHOICE(4019H)

設定　なし
戻り値　HL　メッセージ文字列の先頭アドレス。メッセージは00Hで終わる
ただし、メッセージがない場合(フォーマットのサポートが1種類)HL=0
レジスタ　すべて

DSKFMT(401CH)

```
  設定  A  フォーマットのタイプ(1～9)
           ただし、その意味するところはドライバによって異なる
           フォーマットのサポートが1種類のみのドライバでは意味はない
        D  ドライブ番号(0～)
        HL フォーマット時に使うワークエリアの先頭アドレス
        BC ワークエリアのサイズ
戻り値  成功  Cy=0
        失敗  Cy=1さらにAにエラーコード
         エラーコード         0  Write protected
                              2  Not ready
                              4  Data (CRC) error
                              6  Seek error
                              8  Record not found
                             10  Write fault
                             12  Bad parameter
                             14  Insufficient memory
                             16  Other errors
レジスタ  すべて
```

ディスクを物理的にフォーマットし、また FAT とディレクトリを設定します。このルーチンは内部でメッセージを表示することはありません。

(MOTOR_OFF_ENTRY)(401FH)　(公開)

設定　なし
戻り値　なし
レジスタ　？

ドライブを停止します。

このエントリーは後になって追加されたため初期のドライブではない場合があります。その場合このアドレスは00Hになっていますので、このルーチンをコールする場合は必ずこのアドレスが00Hでないことを確かめてその存在を確認します。

INIHRD

設定　なし
戻り値　なし
レジスタ　AF,BC,DE,HL,IX,IY

システム初期化時にそのインターフェイス ROM に処理が渡ってすぐにハードウェアの初期化を行います。このルーチンが呼ばれる時点ではまだディスクワークエリアは確保されていません。
このルーチンは初期化ルーチンから直接呼ばれるため、アドレスはドライバによって異なります。

DRIVES

　　　設定　Z=0　2ドライブにする(1ドライブなら2ドライブシミュレーションを行う)
　　戻り値　L　接続されているドライブの数
　レジスタ　F,HL,IX,IY

　システム初期化時に、ドライバの処理するドライブ数をシステムに返します。このルーチンが呼ばれる時にはディスクワークエリアが設定されています。
　このルーチンは初期化ルーチンから直接呼ばれるため、アドレスはドライバによって異なります。

INIENV

設定　なし
戻り値　なし
レジスタ　AF,BC,DE,HL,IX,IY

　システム初期化時に、ドライバ用ワークエリアの初期化、タイマ割り込みの設定などを行います。
　このルーチンは初期化ルーチンから直接呼ばれるため、アドレスはドライバによって異なります。

OEMSTATEMENT

　BASICのステートメントを拡張します。Disk BASICが自分の拡張ステートメントかどうか調べてからディスクドライバのこのルーチンに処理が渡ります。
　レジスタ等の設定についてはMSX2テクニカルハンドブックP329を参照して下さい。

３．２　ディスクドライバのタイプ

ここからそれぞれの機種のディスクドライバなどについて話をして行く上での便利のために各々のディスクドライバやDOSカーネルに呼称を割り当てます。これらの名称は独自のもので世間一般に通用するものではありません。しかしながらこれらのデータを利用する際には、混乱を避けるためにも本書に準拠した表記を使用することをお勧めします。

１　DOSカーネルのタイプ

DOSカーネルは機種による大きな違いはありません。しかし、細部の手直しが入っていたり、INITなどのルーチン内のドライブ固有パラメータがある部分で異なっている場合があります。カーネルはドライブ固有パラメータを除けばいくつかのバージョンに分類できます。そこで、バージョン毎に番号を付けてタイプ1などと呼びます。

(現在、全部で幾つのバージョンがあるのか不明なので、タイプの番号は確認された順に付けています。よって、番号の大小とバージョンの新旧は一致しません。)

２　ディスクドライバのタイプ

ディスクドライバは各機種のハードウェアの差異を吸収する部分ですので、メーカーが違えば異なるのはもちろん、同じメーカーでも機種が変わると内容が変わる場合があります。しかし、同一シリーズ内ではI/Oポートの変更はほとんど無く、ルーチンの手直しである場合がほとんどです。

ディスクドライバはあるシリーズ内ではほとんど同じ物であることが多いため、似ているものをまとめて系統と呼びます。メーカーごとに系統ができているのが普通ですから、メーカーの名前を付けて区別します。系統は、ほぼそのディスクドライブのI/OポートとFDCの種別を表すと言っていいでしょう。

系統が同じであっても、ディスクドライバの細部が同じとは限りませんし、時にはI/Oポートが変わっている場合さえあります。そこで、そういった違いを含めてディスクドライバのタイプと呼び、系統名＋番号で表します。また、タイプ名は略してアルファベット2字＋数字で表す場合があります。

３．３　ディスクドライバ資料

１　タイプ別資料の読み方

　ここからそれぞれのディスクドライバの資料を収録しますが、表記を簡便にするため細部については省略してある場合があります。それぞれの項目の内容の説明はここでまとめて紹介しますので資料を見る前に一度目を通して下さい。なお、資料は独自の解析によるもので、すべてのデータが判明しているわけではありませんので注意してください。

タイプ名および該当機種

　タイプ名は前節で紹介したディスクドライバのタイプ名です。括弧内は略記です
　また、タイプ名に続いて、そのタイプに属する機種の機種名および DISK ROM のスロット番号を示します。

各種データ

FDC	使用している FDC のタイプです。括弧内は具体的な FDC の名称です。
ワークエリア長	ディスクドライバが確保するワークエリアのサイズです。
フォーマット用ワークエリア長	フォーマット時にディスクドライバが最低限必要とするワークエリアのサイズです。フォーマットルーチンをコールするときにこれ以上のメモリが確保できないとエラーを返します。
対応メディア	ディスクドライバがサポートしているメディアです。メディア ID と、フォーマット可能な場合は、フォーマットで指定する番号とメディアタイプの関係を示します。

特徴

　そのドライブに関して、特に説明すべき点を解説します。

ドライバ内サブルーチンアドレス

どのディスクドライバでも共通に存在するルーチンの先頭アドレスを示します。詳しくは３．１．３ディスクドライバの提供するルーチンを参照してください。なお次のような略記をしています。

```
MOTOR_OFF_ENTRY      (略)MTROFF
OEMSTATEMENT         (略)OEMSTA
```

ディスクドライバ関係定数

ディスクドライバに関する定数の内、カーネルの初期化ルーチン部分に含まれているデータを示します。

MYSIZE　ドライバ用ワークエリアのサイズです。
SECLEN　そのドライバがサポートするメディアのうち最大のセクタサイズです。
DEFDPB　FAT のサイズが最大のメディアの DPB データのアドレスです。

I/O ポート

FDC 及びディスクドライブの I/O ポートを説明します。

ここで説明する I/O ポートは Z80 の I/O ポートを使用していません。DISK ROM のスロット上のメモリ空間にマップされています。そのスロットに切り替えてから該当アドレスをアクセスします。

ワークエリア

ディスクドライバ用ワークエリアの内容について説明します。

アドレス表示はすべてワークエリアの先頭アドレスからのオフセット値です。

フォーマット用ワークエリア

フォーマット時にディスクドライバが使用するワークエリアについて説明します。アドレス表示はすべてワークエリアの先頭アドレスからのオフセット値です。ワークエリアのアドレスはフォーマットルーチンを呼び出すプログラムが指定します。

2 National

タイプ名および該当機種

タイプ名		メーカー	機種名	スロット
National1	NA1	National	FS-4600	
National2	NA2		FS-5500F1/2	3-3
			FS-4700F	3-3

各種データ

FDC	ワークエリア長	フォーマット用ワークエリア長
8876系(MB8877A)	10バイト	6KB

対応メディア　F8H～FFH　1:FCH 2:FDH 3:F8H 4:F9H

特徴

　このシステムは、メディアの交換をハード的に検出できないため、一定時間アクセスが無いとメディアが交換されたと見なします。時間の測定は、タイマ割り込みで処理されます。また、ドライブの停止もタイマ割り込みで一定時間後に行われます。

　モータースイッチは、ドライブセレクト信号に無関係に両ドライブに対して有効です。ドライブセレクト信号は、1でそのドライブが有効になり、アクセスランプはドライブセレクト信号に連動します。FDCのREADY入力は常に1のようです。

　独自の拡張ステートメントが存在し、CALL VERIFY ON/OFFでベリファイの有無を設定できます。これはDOSの同名コマンドと同じ働きをします。

ドライバ内サブルーチンアドレス

	DSKIO	DSKCHG	GETDPB	CHOICE	DSKFMT	MTROFF	OEMSTA	INIHRD	DRIVES	INIENV
NA1	7495H	7868H	78A8H	78C3H	7BB9H	77E5H	7919H	77D9H	7806H	7830H
NA2	7495H	785AH	789AH	78B5H	7BBFH	77D7H	790BH	77C6H	77F8H	7822H

ディスクドライバ内定数

```
      MYSIZE SECLEN DEFDPB
NA1-2 000AH  0200H  7416H
```

I/O ポート

BFB8H～BFBCH にもマップされています。

```
7FB8H R/W ステータスレジスタ/コマンドレジスタ
7FB9H R/W トラックレジスタ
7FBAH R/W セクタレジスタ
7FBBH R/W データレジスタ
7FBCH R   FDC ステータス      / IRQ /-DRQ /  -  /  -  /  -  /  -  /  -  /  -  /
      W   FD コントロール     /  0  /  0  /  0  /  0  / MOT /SIDE / D1  / D0  /

  IRQ  FDC の IRQ 出力                 -DRQ FDC の DRQ 出力の反転
  MOT  1 でモーター作動                SIDE サイドセレクト
  D1   ドライブセレクト 1              D0   ドライブセレクト 0
```

ワークエリア

```
+00H,1  モーターの状態(00H:停止  01H～FEH:待機  FFH:作動)
        待機状態のときはタイマ割り込みで値が減じられ 0 になるとモーターを止める
+01H,1  ドライブ 0 のメディア交換判定用タイマー
        タイマ割り込みで値が減じられ 0 になるとメディアが交換されたと見なす
+02H,1  ドライブ 1 のメディア交換判定用タイマー
+03H,1  最後にアクセスした物理ドライブ(FDC が現在トラック番号を保持する)
+04H,1  ドライブ 1 を選択時、ドライブ 0 のトラック番号を保存する
+05H,1  ドライブ 0 を選択時、ドライブ 1 のトラック番号を保存する
+06H,1  2 ドライブシミュレーション時、現在の仮想ドライブ番号
+07H,1  内部で使用する。ベリファイかどうかのフラグ
+08H,1  内部で使用する。シーク後にウェイトを入れるかどうか
+09H,1  物理ドライブの数。1 なら 2 ドライブシミュレーションが有効
```

フォーマット用ワークエリア

```
+00H,2  ブートセクタ用データの格納アドレス
+02H,1  メディア ID
+03H,2  セクタの数
+05H,2  トラックデータを展開するワークエリアのアドレス(+20H)
+07H,1  ドライブの状態に関する情報
+08H,1  エラー発生時のリトライ回数
+09H,1  物理ドライブ番号
+20H,   フォーマットデータを展開するエリア
```

3　Panasonic

タイプ名および該当機種

タイプ名		メーカー	機種名	スロット
Panasonic 1	PA1	Panasonic	FS-FD1A	(拡張ドライブ)
Panasonic 2	PA2		FS-A1F	3-2
Panasonic 3	PA3		FS-A1FX	3-2
			FS-A1WX	3-2
			FS-A1WSX	3-2
Panasonic 4	PA4		FS-A1ST	3-2
Panasonic 5	PA5		FS-A1ST	3-2
			FS-A1GT	3-2
Panasonic 6	PA6		FS-A1ST	3-2

各種データ

FDC	ワークエリア長	フォーマット用ワークエリア長
765系(TC8566AF)	26バイト	5KB

対応メディア　　F8H～FFH　　1:F8H 2:F9H

特徴

FDCには東芝のTC8566AFを使っています。TC8566AFは本来のFDC部分に加えてドライブコントロール用の周辺回路までをまとめたチップで、ドライブコントロール用の拡張レジスタを2本持っています。

FS-A1STでは生産時期によってROMバージョンの異なる物が3タイプ以上はあることが判明しています。

A1ST以降、DOS2を本体内に搭載していますが、DOS2の場合とDOS1の場合でディスクドライバが違います。DOS2でのディスクドライバはメディアの交換の検出等の機能も持っています。また、ST以降I/Oアドレスも変更されました。

ドライバ内サブルーチンアドレス(PA4～PA5はDOS1時)

	DSKIO	DSKCHG	GETDPB	CHOICE	DSKFMT	MTROFF	OEMSTA	INIHRD	DRIVES	INIENV
PA1	7495H	7809H	7848H	7863H	7C81H	7770H	78A0H	7733H	77A1H	77CCH
PA2	7495H	77E4H	7823H	783EH	7C5CH	774CH	787BH	772AH	7783H	77AAH
PA3	7495H	77F3H	7833H	784EH	7C80H	775AH	788BH	7738H	7792H	77B9H
PA4	7495H	77CAH	780AH	7825H	7C57H	7746H	7862H	7724H	777AH	77A1H
PA5	7495H	77CBH	780BH	7826H	7C58H	7747H	7863H	7725H	777BH	77A2H
PA6	7495H	77CAH	780AH	7825H	7C57H	7746H	7862H	7724H	777AH	77A1H

ディスクドライバ内定数

```
       MYSIZE SECLEN DEFDPB
PA1-6 001AH  0200H  7416H
```

I/O ポート

FDC は TC8566AF を使用しており、TC8566AF の拡張レジスタをドライブコントロールのために使用しています。I/O のアドレスが PA1〜PA3 と PA4〜PA5 では異なります。

I/O は同じスロットの BFF8H〜BFFBH(PA1〜PA3)、BFF1H〜BFF5H(PA4〜PA5)にもそれぞれ同じ内容がマップされています。PA4〜PA5 では、メディアの交換検出用の I/O ポートが新設されています。

```
(PA1〜PA3)
7FF8H W   FDC 拡張レジスタ 0  /  0  /  0  /MEN1 /MEN0 /  0  /-FRST/  0  / DSA /
7FF9H W   FDC 拡張レジスタ 1  /  0  /  0  / C4E / C4  /SBME / SBM / TCE /FDCTC/
7FFAH R   ステータスレジスタ
7FFBH R/W データレジスタ

(PA4〜PA5)
7FF1H R   FDD ステータス      /  -  /  -  / MC1 / MC0 /  -  /  -  /  -  /  -  /
7FF2H W   FDC 拡張レジスタ 0  /  0  /  0  /MEN1 /MEN0 /  0  /-FRST/  0  / DSA /
7FF3H W   FDC 拡張レジスタ 1  /  0  /  0  / C4E / C4  /SBME / SBM / TCE /FDCTC/
7FF4H R   ステータスレジスタ
7FF5H R/W データレジスタ

  MEN1,MEN0      モーターイネーブル 1/0  モーターを 1 で ON
  -FRST          FDC リセット信号の反転  0 を書き込むとリセット
  DSA            ドライブセレクト  0 でドライブ 0
  C4E            C4 への書き込みを有効にする
  C4             強制 READY  ドライブが NOT READY 時でも FDC の入力は READY にする
  SBME           SBM への書き込みを有効にする
  SBM            スタンバイモード  0 の時はスタンバイ状態にはならない
  TCE            FDCTC への書き込みを有効にする
  FDCTC          TC(ターミナルカウント)
  MC1,MC0        メディア交換の検出用
```

ワークエリア

+00H,1　モーターの状態(00H:OFF FFH:ON)
+01H,1　メディアが有効か。ドライブ0
+02H,1　　　　　　　　　　ドライブ1
+03H,1　2ドライブシミュレーション時の現在のドライブ番号(0～1)
+04H,1
+05H,1　リード／ライト別のフラグ(b0)
+06H,1　モーターの状態(I/O書き込み用データ)
+07H,1　物理ドライブ数(0:1ドライブ　1:2ドライブ)
+08H,1
+09H,1
+0AH,9　FDCコマンド用バッファ
+13H,7　リザルトステータス用バッファ

フォーマット用ワークエリア

+00H,2　ブートセクタ用データのアドレス
+02H,1　メディアID
+03H,2　セクタ数
+05H,2
+07H,1　論理ドライブ番号
+08H,1　物理ドライブ番号
+09H,1　フォーマットタイプ-1
+0AH,1
+0BH,36 フォーマットパラメータ用バッファ(4バイト*9セクタ分)
+2FH,
+39H,　FDCコマンド用バッファ
+200H,　リード用バッファ

4　SONY

タイプ名および該当機種

タイプ名		メーカー	機種名	スロット
SONY 1	S01	SONY	HB-F900	
SONY 2	S02		HBD-F1	（拡張ドライブ）
			HB-F1XD	
			HB-F1XDmkII	
			HB-F1XDJ	
			HB-F1XV	
	S03			
SONY 4	S04		FS-FD1	（拡張ドライブ）

各種データ

FDC	ワークエリア長	フォーマット用ワークエリア長
8876系(TMS2793)	9バイト	使用しません

対応メディア	F8H～FBH	1:F8H 2:F9H

特徴

ドライブはI/Oポートを切り替えてからも約4秒間動作し続け、その後自動的に停止します。このドライバはフォーマット時にユーザーのワークエリアを使用しません。

FS-FD1(SO4)はPanasonicブランドですが、ソニーによるOEM生産品です。フォーマット時のメッセージがSONYのものとは違います。

SO2のディスクドライバには2DD8セクタの時のパラメータが誤っているというバグが発見されています。そのパラメータを修正したものがSO3です。

ドライバ内サブルーチンアドレス

	DSKIO	DSKCHG	GETDPB	CHOICE	DSKFMT	MTROFF	OEMSTA	INIHRD	DRIVES	INIENV
S01	7512H	751EH	7521H	7524H	7527H	752DH	752AH	7515H	7518H	751BH
S02-3	751EH	78BAH	7943H	795DH	79A1H	7DE2H	7DE0H	7827H	7867H	78A7H
S04	751EH	78BAH	7943H	795DH	79A0H	7DE2H	7DE0H	7827H	7867H	78A7H

ディスクドライバ内定数

	MYSIZE	SECLEN	DEFDPB
S01	0009H	0200H	74DBH
S02-4	0009H	0200H	74E7H

I/O ポート

ポート	R/W	名称	b7	b6	b5	b4	b3	b2	b1	b0
7FF8H	R	ステータスレジスタ								
	W	コマンドレジスタ								
7FF9H	R/W	トラックレジスタ								
7FFAH	R/W	セクタレジスタ								
7FFBH	R/W	データレジスタ								
7FFCH	R	FDD ステータス 1	-	-	-	-	-	TM	IU	SIDE
	W	FDD コントロール 1	0	0	0	0	0	0	0	SIDE
7FFDH	R	FDD ステータス 2	MOT	-	-	-	-	TM	IUC	DS
	W	FDD コントロール 2	MOT	0	0	0	0	TE	IUC	DS
7FFFH	R	FDC ステータス	-DRQ	-IRQ	-	-	-	-	-	-

TM　ディスクチェンジチェックタイマー
TE　タイマーイネーブル
IU　IN USE 信号の状態
IUC　IN USE 信号制御とタイマーセット
SIDE　サイドセレクト
DS　ドライブセレクト
MOT　モーターの状態　0 を書き込んでもモーターはしばらく作動し続け、その間はこのビットを読むと 1 が読み出されます。
-DRQ　FDC の DRQ 信号の反転　　-IRQ　FDC の IRQ 信号の反転

ワークエリア

+00H,1
+01H,1
+02H,1
+03H,1　FDC が現在トラック番号を保持しているドライブの番号
+04H,1　ドライブ 1 を選択時、ドライブ 0 のトラック番号
+05H,1　ドライブ 0 を選択時、ドライブ 1 のトラック番号
+06H,1　2 ドライブシミュレーション時、現在の仮想ドライブ番号
+07H,1
+08H,1　物理ドライブの数

5　SANYO

タイプ名および該当機種

タイプ名		メーカー	機種名	スロット
SANYO 1	SA1	SANYO	PHC-70FD	3-2
			PHC-70FD2	3-2

各種データ

FDC	ワークエリア長	フォーマット用ワークエリア長
765系(TC8566AF)	26バイト	5KB

対応メディア　　F8H～FFH　　1:F8H 2:F9H

特徴

この機種はPanasonicの機種とI/Oポートが同じです。ディスクドライバのプログラムも同じなのですが、メッセージを書き換えたり、ルーチンどうしの順番を入れ替えたりしてあり、一見違うように見えます。

FDCには東芝のTC8566AFを使っています。TC8566AFは本来のFDC部分に加えてドライブコントロール用の周辺回路までをまとめたチップで、ドライブコントロール用の拡張レジスタを2本持っています。

ドライバ内サブルーチンアドレス

	DSKIO	DSKCHG	GETDPB	CHOICE	DSKFMT	MTROFF	OEMSTA	INIHRD	DRIVES	INIENV
SA1	77F5H	7B63H	7BA3H	7BBEH	7405H	7ABAH	7BFBH	7A98H	7B02H	7B29H

ディスクドライバ内定数

	MYSIZE	SECLEN	DEFDPB
SA1	001AH	0200H	7776H

I/O ポート

```
7FF8H W   FDC拡張レジスタ0  /  0  /  0  /MEN1 /MEN0 /  0  /-FRST/  0  / DSA /
7FF9H W   FDC拡張レジスタ1  /  0  /  0  / C4E / C4  /SBME / SBM / TCE /FDCTC/
7FFAH R   ステータスレジスタ
7FFBH R/W データレジスタ
```

```
MEN1,MEN0      モーターイネーブル1/0  1でON
-FRST          FDCリセット信号の反転  0を書き込むとリセット
DSA            ドライブセレクト  0でドライブ0
C4E            C4への書き込みを有効にする
C4             強制READY  ドライブがNOT READY時でもFDCにはREADYが入力される
SBME           SBMへの書き込みを有効にする
SBM            スタンバイモード  0の時はスタンバイ状態にはならない
TCE            FDCTCへの書き込みを有効にする
FDCTC          TC(ターミナルカウント)
```

ワークエリア

```
+00H,1  モーターの状態(00H:OFF FFH:ON)
+01H,1  メディアが有効か。ドライブ0
+02H,1                    ドライブ1
+03H,1  2ドライブシミュレーション時の現在のドライブ番号(0~1)
+04H,1
+05H,1  リード/ライト別のフラグ(b0)
+06H,1  モーターの状態(I/O書き込み用データ)
+07H,1  物理ドライブ数(0:1ドライブ  1:2ドライブ)
+08H,1
+09H,1
+0AH,9  FDCコマンド用バッファ
+13H,7  リザルトステータス用バッファ
```

フォーマット用ワークエリア

```
+00H,2  ブートセクタ用データのアドレス
+02H,1  メディアID
+03H,2  セクタ数
+05H,2
+07H,1  論理ドライブ番号
+08H,1  物理ドライブ番号
+09H,1  フォーマットタイプ-1
+0AH,1
+0BH,36 フォーマットパラメータ用バッファ(4バイト*9セクタ分)
+2FH,
+39H,   FDCコマンド用バッファ
+200H,  リード用バッファ
```

6　Victor

タイプ名および該当機種

タイプ名		メーカー	機種名	スロット
Victor 1	VI1	Victor	HC-90 HC-95	

各種データ

FDC	ワークエリア長	フォーマット用ワークエリア長
8876系	8バイト	80Hバイト

対応メディア　　F8H～FFH　　1:F8H 2:F9H

特徴

このHC-90とHC-95はZ80の上位互換CPUであるHD64180を搭載したMSX2で、HD64180を使用するモードではZ80よりも高速動作するという機能がありました。その影響で、ディスクドライバにはウェイト調整のようなプログラムが組み込んであります。またHD64180ではZ80にはない独自の拡張命令が存在し、ディスクドライバ内部でも使われています。

ドライバ内サブルーチンアドレス

	DSKIO	DSKCHG	GETDPB	CHOICE	DSKFMT	MTROFF	OEMSTA	INIHRD	DRIVES	INIENV
VI1	7495H	7A5DH	7A9CH	7AB7H	7AF3H	7997H	7F24H	7979H	79D4H	7A21H

ディスクドライバ内定数

	MYSIZE	SECLEN	DEFDPB
VI1	0008H	0200H	7416H

I/O ポート

```
7FF8H R   ステータスレジスタ
      W   コマンドレジスタ
7FF9H R/W トラックレジスタ
7FFAH R/W セクタレジスタ
7FFBH R/W データレジスタ
7FFCH R   コントロール1    / IRQ / DRQ /  -  /  -  /SIDE / DS  /  -  /  -  /
      W                   /     /     /     /-MOT /SIDE / DS  /  M1 /  M0 /
7FFDH R   コントロール2    /  ?  / CPU /  -  /  -  /  -  /  -  /  -  /  -  /

  IRQ   FDCのIRQ信号                DRQ   FDCのDRQ信号
  -MOT  ドライブのモーター(0でON)
  SIDE  サイドセレクト
  DS    ドライブセレクト
  CPU   HD64180モードだと1
```

7　TOSHIBA

タイプ名および該当機種

タイプ名		メーカー	機種名	スロット
TOSHIBA 1	T01	TOSHIBA	HX-34	

各種データ

FDC	ワークエリア長	フォーマット用ワークエリア長
8876系	10バイト	使用しません

対応メディア　　F8H〜FCH　　1:F8H 2:F9H

ドライバ内サブルーチンアドレス

	DSKIO	DSKCHG	GETDPB	CHOICE	DSKFMT	MTROFF	OEMSTA	INIHRD	DRIVES	INIENV
T01	753FH	7936H	79B7H	79D6H	7A0FH	7D4EH	74E8H	74EAH	7511H	752FH

ディスクドライバ内定数

	MYSIZE	SECLEN	DEFDPB
T01	000AH	0200H	7469H

I/O ポート

```
 7FF0H R     ステータスレジスタ
       W     コマンドレジスタ
 7FF1H R/W   トラックレジスタ
 7FF2H R/W   セクタレジスタ
 7FF3H R/W   データレジスタ
 7FF4H R/W   FDコントロール1  /     /     /     /     /     /     / MOT /SIDE /
 7FF5H R/W   FDコントロール2  /     /     /     /     /     /     /     /  D  /
 7FF6H R/W   FDコントロール3  /     /     /     /     /     /     /     / -DC /
 7FF7H R     FDCステータス    /-DRQ / IRQ /  -  /  -  /  -  /  -  /  -  /  -  /

   MOT   1でモーターON
   SIDE  サイドセレクト信号
   -DRQ  FDCのDRQ信号の反転
   IRQ   FDCのIRQ信号
   D     ドライブセレクト
   -DC   ディスクチェンジ信号の反転
```

ワークエリア

```
+00H,1  0:ドライブ0モーター停止
+01H,1  0:ドライブ1モーター停止
```

8　CANON

タイプ名および該当機種

タイプ名		メーカー	機種名	スロット
CANON 1	CA1	CANON	V-30F	3-1

各種データ

FDC	ワークエリア長	フォーマット用ワークエリア長
8876系	7バイト	使用しません

対応メディア　　F8H～FBH　　1:F8H 2:F9H

ドライバ内サブルーチンアドレス

	DSKIO	DSKCHG	GETDPB	CHOICE	DSKFMT	MTROFF	OEMSTA	INIHRD	DRIVES	INIENV
CA1	744DH	77ABH	7816H	7842H	7BC1H	7748H	782DH	7728H	776BH	77A0H

ディスクドライバ内定数

	MYSIZE	SECLEN	DEFDPB
CA1	0007H	0200H	7416H

I/Oポート

```
7FF0H R     ステータスレジスタ
      W     コマンドレジスタ
7FF1H R/W   トラックレジスタ
7FF2H R/W   セクタレジスタ
7FF3H R/W   データレジスタ
7FF4H R/W                      /     /     /     /     /     /     / MOT /SIDE /
7FF5H R/W
7FF6H R/W
7FF7H R     FDCステータス    /-DRQ / IRQ /  -  /  -  /  -  /  -  /  -  /  -  /
```

MOT　1でモーターON
SIDE　サイドセレクト信号
-DRQ　FDCのDRQ信号の反転
IRQ　FDCのIRQ信号

9　MITSUBISHI

タイプ名および該当機種

タイプ名	メーカー	機種名	スロット
MITSUBISHI 1 MI1	MITSUBISHI	ML-G30	

各種データ

FDC	ワークエリア長	フォーマット用ワークエリア長
8876系	8バイト	15バイト

対応メディア　　F8H～FFH　　1:F8H　2:F9H

ドライバ内サブルーチンアドレス

	DSKIO	DSKCHG	GETDPB	CHOICE	DSKFMT	MTROFF	OEMSTA	INIHRD	DRIVES	INIENV
MI1	7495H	783FH	787EH	7899H	78DEH	779AH	7CF8H	7789H	77D6H	780AH

ディスクドライバ内定数

	MYSIZE	SECLEN	DEFDPB
MI1	0008H	0200H	7416H

I/O ポート

```
7FF8H R    ステータスレジスタ
      W    コマンドレジスタ
7FF9H R/W トラックレジスタ
7FFAH R/W セクタレジスタ
7FFBH R/W データレジスタ
7FFCH R    FDD ステータス 1    /  -  /  -  /  -  /  -  /  -  /  -  /  -  /SIDE /
      W    FDD コントロール 1  /  0  /  0  /  0  /  0  /  0  /  0  /  0  /SIDE /
7FFDH R    FDD ステータス 2    / MOT /  -  /  -  /  -  /  -  /  -  / IUC / DS  /
      W    FDD コントロール 2  / MOT /  0  /  0  /  0  /  0  /  0  / IUC / DS  /
7FFFH R    FDC ステータス      /-DRQ /-IRQ /  -  /  -  /  -  /  -  /  -  /  -  /
```

IUC　IN USE 信号制御とタイマーセット
SIDE　サイドセレクト
DS　ドライブセレクト
MOT　モーターの状態　0 を書き込んでもモーターはしばらく作動し続け、その間はこのビットを読むと 1 が読み出されます。
-DRQ　FDC の DRQ 信号の反転
-IRQ　FDC の IRQ 信号の反転

１０　YAMAHA

タイプ名および該当機種

タイプ名		メーカー	機種名	スロット
YAMAHA 1	YA1	YAMAHA	FD-05	(拡張ドライブ)

各種データ

FDC	ワークエリア長	フォーマット用ワークエリア長
8876系	10バイト	6KB

対応メディア　F8H～F9H　1:F9H 2:F8H

ドライバ内サブルーチンアドレス

	DSKIO	DSKCHG	GETDPB	CHOICE	DSKFMT	MTROFF	OEMSTA	INIHRD	DRIVES	INIENV
YA1	74F1H	7865H	78A4H	78E4H	7911H	77D0H	7C14H	77BFH	77F0H	7832H

ディスクドライバ内定数

	MYSIZE	SECLEN	DEFDPB
YA1	000AH	0200H	74D5H

I/O ポート

```
7FF8H R   ステータスレジスタ
      W   コマンドレジスタ
7FF9H R/W トラックレジスタ
7FFAH R/W セクタレジスタ
7FFBH R/W データレジスタ
7FFCH R   FDD ステータス 1     /  -  /  -  /  -  /  -  /  -  /  -  /  -  /SIDE /
      W   FDD コントロール 1   /  0  /  0  /  0  /  0  /  0  /  0  /  0  /SIDE /
7FFDH R   FDD ステータス 2     / MOT /  -  /  -  /  -  /  -  /  -  / IUC / DS  /
      W   FDD コントロール 2   / MOT /  ?  /  ?  /  ?  /  ?  /  ?  / IUC / DS  /
7FFEH W   FDD コントロール 3   /  ?  /  ?  /  ?  /  ?  /  ?  /  ?  /  ?  /  ?  /
7FFFH R   FDC ステータス      /-DRQ /-IRQ / TE  /  -  /  -  /  -  /  -  /  -  /

  TE     タイマーイネーブル
  IUC    IN USE 信号制御とタイマーセット
  SIDE   サイドセレクト
  DS     ドライブセレクト
  MOT    モーターの状態  0 を書き込んでもモーターはしばらく作動し続け、その間
                         はこのビットを読むと 1 が読み出されます。
  -DRQ   FDC の DRQ 信号の反転
  -IRQ   FDC の IRQ 信号の反転
```

１１　ASCII

タイプ名および該当機種

```
    タイプ名          メーカー    機種名
ASCII 1       AS1   ASCII       HDインターフェイス
```

各種データ

ワークエリア長
　34バイト

ドライバ内サブルーチンアドレス

```
      DSKIO  DSKCHG GETDPB CHOICE DSKFMT MTROFF OEMSTA INIHRD DRIVES INIENV
AS1    74A8H  7994H  79ACH  7A61H  7A6FH  7949H  7B26H  7717H  77F6H  78E7H
```

ディスクドライバ内定数

```
      MYSIZE SECLEN DEFDPB
AS1    0022H  0200H  7495H
```

I/Oポート

BFFCH～BFFDHにもマップされています。

```
7FFC R/W データポート
7FFD W   コマンドポート
          ビット0 予約済み(通常は0を書き込む)
          ビット1 予約済み(通常は0を書き込む)
          ビット2 ATTENTION(1ならTRUE)
          ビット3 SELECT(1ならTRUE)
     R   ステータスポート
          ビット0 REQ(1ならTRUE)
          ビット2 ATTENTION(1ならTRUE)
          ビット3 SELECT(1ならTRUE)
          ビット4 BUSY(0ならBUSY)
          ビット5 MSG(0ならMESSAGE)
          ビット6 COMMAND/DATA(0ならCOMMAND)
          ビット7 INPUT/OUTPUT(0ならHOST<-SCSI)
```

第 4 章　FDC

４．１　FDC

CPU がフロッピーディスクにアクセスする時に CPU とディスクドライブの間を結んでいるのが FDC(フロッピーディスクコントローラ)です。MSX では FDC の種類や FDC との I/O レベルでのインターフェイスは定められていないため、同じ 3.5 インチ 2DD のドライブを使っている機種でも FDC や I/O ポート等は異なっています。

FDC には代表的な 2 つの系列があり、765 系、8876 系などと呼ばれています。

４．２　μPD765 系

１　μPD765

μPD765 という FDC は 1978 年に日本電気が開発した FDC です。現在、この系統の FDC が主流になっています。4 台までの FDD を接続することを想定した設計になっていて、ドライブコントロールがしやすくなっていることや、トラックフォーマットのデータを内蔵していてホストの負担が軽くなっていることなどの特徴があります。

μPD765 の後に、いくつもの改良品が出されており、また各社からセカンドソースが供給されています。これらの FDC をまとめて μPD765 ファミリとか単に 765 系とかいいます。

２　レジスタ

765 系の FDC はステータスレジスタとデータレジスタの 2 本のレジスタを持っています。

データレジスタは、8 ビットのレジスタで、FDC へのコマンドの転送やデータの読み出しと書き込みに使用します。

ステータスレジスタはその名の通り FDC の状態を示しています。FDC と CPU の間でデータ転送をする際にはステータスレジスタを見ながらデータ転送をすることになります。

ステータスレジスタは 8 ビットのレジスタで各ビットがそれぞれ意味を持っています。各ビットの意味は下に示します。ステータスレジスタは読み込み専用で書き込みは禁止となっています。

ステータスレジスタの内容

	名称	略称	機能
b0	FD0 Busy	D0B	ドライブ0がSeekコマンドによるシークを実行中であるか、シーク終了の割り込み要求を保留中である。
b1	FD1 Busy	D1B	ドライブ1について、D0Bと同じ。
b2	FD2 Busy	D2B	ドライブ2について、D0Bと同じ。
b3	FD3 Busy	D3B	ドライブ3について、D0Bと同じ。
b4	FDC Busy	CB	FDCがコマンド・リザルトの各フェーズか、リード／ライトコマンドのエグゼキューションフェーズを実行中である。このビットがセットされているときは他のコマンドは受け付けられない。
b5	Non-DMA Mode	NDM	FDCがNon-DMAモードでデータ転送中であり、メインシステムに対してサービスを要求している。0に変化する時はエグゼキューションフェーズの終了です。
b6	Data Input/Output	DIO	データレジスタの転送方向を示す。 0:メインシステムからFDC 1:FDCからメインシステム
b7	Request for Master	RQM	メインシステムに対するデータ転送要求。 DIO=0のとき メインシステムがデータレジスタにデータをセットしたとき0に、FDCがそのデータを引き取ると1になる。つまり、1になったらFDCにデータを転送できる。 DIO=1のとき FDCがデータレジスタにデータをセットしたとき1に、メインシステムがそのデータを受け取ると0になる。つまり、1になったらFDCからデータを受け取れる。

3　ドライブコントロール

FDC自体にはドライブのモーターをコントロールするなどの機能はないので、これらはFDCとは別に外部回路があります。外部回路は機種によって異なります。

外部回路のコントロールのためにFDCのレジスタの他にいくつかのI/Oポートを使用します。

4　FDC のコントロール

FDC にある動作をさせる場合には、FDC はコマンドという形でシステムの要求を実行します。コマンドの実行は 3 つの段階にわけて行われ、そのそれぞれをフェーズと言います。FDC にコマンドを実行させるには FDC に対してコマンドを送ります。これをコマンドフェーズといいます。そして、データを転送するのがエグゼキューションフェーズです。コマンドが終了すると、FDC が実行の成否などをホストに知らせます。これがリザルトフェーズです。どのフェーズにおいても、コマンドやデータの転送はステータスレジスタを見ながらデータレジスタを介しておこなわれます。

コマンド実行の手順を説明します。

まず、コマンドフェーズでは、FDC が Busy でないことを確認してから実行させたいコマンドとパラメータ(合計 1〜9 バイト)を FDC に転送します。転送の前には、必ずステータスレジスタを読んで FDC がデータ転送の準備ができているかを確認します。

次は、エグゼキューションフェーズです。ここでは実際のデータを転送します。ここでいう実際のデータとは、セクタの内容やフォーマットの際にシステムが指定するデータ等です。エグゼキューションフェーズではコマンドの種類により、FDC からシステムにデータ転送をする場合、システムから FDC にデータ転送をする場合があります。

コマンドによっては転送すべきデータがないので、エグゼキューションフェーズがありません。

コマンドが終了するとリザルトフェーズで FDC が実行結果をシステムに知らせます。
リザルトフェーズでは 1〜7 バイトのデータを CPU が受け取ります。コマンドによってはリザルトフェーズのないものもあります。

FDC のコマンド実行は以上の 3 つのフェーズで完了します。

FDC の RESET 入力を 1 にすることで FDC にリセットを掛けることもできます。

5　リザルトステータス

リザルトフェーズでFDCから返されるデータにリザルトステータスがあります。リザルトステータスはFDC内部にリザルトステータスレジスタという4本のレジスタがあり、コマンドによって異なりますがこのうちのいくつかが返されます。

リザルトステータスレジスタの内容を以下に説明します。

```
リザルトステータスレジスタ0（ST0)
      名称              略称    機能
b0  Unit Select0     US0  INT要求時のデバイス番号を返す。
b1  Unit Select1     US1       〃
b2  Head Address     HD   INT要求時のヘッドの状態
                          Sense Interrupt Status実行時は常に0
b3  Not Ready        NR   指定したデバイスがReady状態でない時セットする。
b4  Equipment Check  EC   デバイスからFault信号を受け取った時、または
                          Recalibrateコマンドで Track0信号が一定時間内に検出
                          できなかった時にセットする。
b5  Seek End         SE   Seek又はRecalibrateコマンドによるシーク動作が正常
                          終了または異常終了したことを示す。
b6  Interrupt Code   IC   INT要求が何によるかを示す。
b7      〃                 b7 b6  00:コマンドの正常終了(NT)
                                 01:コマンドの異常終了(AT)
                                 10:起動したコマンドがInvalidだったため、
                                    マンドを実行しなかったことを示す。(IC)
                                 11:デバイスに状態遷移があったことを示す(AI)
```

```
リザルトステータスレジスタ1（ST1)
      名称              略称    機能
b0  Missing Address  MA  (1)IDをアクセスするコマンドでインデックスパルスを
    Mark                  2回検出するまでにIDAMが見つからない時セットする。
                         (2)IDAMが見つかったあと、DAM又はDDAMが見つからない
                          とセットする。この時、ST2のMDビットもセットする。
b1  Not Writable     NW  ライト系コマンドで、ライトプロテクト信号を検出した。
b2  No Data          ND  (1)次の5種類のコマンド実行時にID情報で指定したセク
                          タがトラック上で検出できないとセットする。
                          Read Data/Read Deleted Data/Write Data
                          Write Deleted Data/Scan
                         (2)Read IDコマンド実行トラック上にCRCエラーのない
                          IDが見つからないとセットする。
                         (3)Read Diagnosticコマンド実行時、セクタIDと指定ID
                          の内容が一致しないとセットする。
b3   ---                 0
b4  Over Run         OR  データ転送時にメインシステムのサービスが規定時間内
                         に行われないとセットする。
b5  Data Error       DE  ディスク上のIDまたはデータのCRCエラーを検出した(Read
                         IDを除く)。ID、データの区別はST2のDDビットによる。
b6   ---                 0
b7  End of Cylinder  EN  EOTで指定した最終セクタを越えてリード／ライトを続
                         ようとした時セットする。
```

リザルトステータスレジスタ2（ST2）

	名称	略称	機能
b0	Missing Address Mark in Data Field	MD	ST1のMAビットの(2)の時セットされる。
b1	Bad Cylinder	BC	ST1のNDビットに付帯してIDのCバイトが一致せずFFHであるとセットする。（Read Diagnosticを除く）
b2	Scan Not Satisfied	SN	Scanコマンドで最終セクタまで条件を満足しないとセットする。
b3	Scan Equal Hit	SH	ScanコマンドでEqual条件を満足するとセットする。
b4	No Cylinder	NC	ST1のNDビットに付帯してIDのCバイトが一致しないでFFHでもない時セットする。（Read Diagnosticを除く）
b5	Data Error in Data Field	DD	データのCRCエラーを検出するとセットする。
b6	Control Mark	CM	Read Data、Read Diagnostic、Scanコマンド実行時にDDAMを検出した時、またはRead Deleted Data実行時にDAMを検出した時にセットする。
b7	---		0

リザルトステータスレジスタ3（ST3）

	名称	略称	機能
b0	Unit Select0	US0	デバイスへのUnit Select0信号の状態
b1	Unit Select1	US1	デバイスへのUnit Select1信号の状態
b2	Head Address	HD	デバイスへのSide Select信号の状態
b3	Two Side	TS	デバイスからのTwo Side信号の状態
b4	Track0	T0	デバイスからのTrack0信号の状態
b5	Ready	RY	デバイスからのReady信号の状態
b6	Write Protected	WP	デバイスからのWrite Protected信号の状態
b7	Fault	FT	デバイスからのFault信号の状態

6　FDCのコマンド

　FDCのコマンドについて説明します。普通使用することのないと思われるコマンドについては説明を簡略化してあります。また、通常のMSXでの使用を前提に記述してありますので、他の用途に使用する場合は参考文献などを参照して下さい。

```
765のコマンド説明で使われる略称
MT      Multitrack     マルチトラック操作(両面のトラックを続けてアクセス)を指
                       定します。
MF      MFM Mode       1でMFMモード、0でFMモードを指定します。FMモードは単密
                       度ディスクの書き込みで使用される書き込み方式です。
SK      Skip           DDAMをスキップすることを指定します。
HD      Head           ヘッド番号を指定します。
US1,US0 Unit Select    ドライブ番号を指定します。
EOT     End of Track   トラック上の最終セクタを指定します。
GPL     Gap Length     VFO SYNCを除いたGAP3の長さを指定します。
```

Read Data

　セクタのリードです。コマンドで指定したID情報(C、H、R、N)により、セクタのデータを読み取ります。

　HDはヘッドの選択です。0でSIDE0になります。US1/US0はドライブの指定です。MSXでは2ドライブまでしかシステムがサポートしませんので、US1は常に0になります。

MT=0、MF=1、SK=0、EOT=9、GSL=54H、DTL=0と指定して下さい。

(2DD/1DDディスクのアクセスはこれで問題ありません。)

　セクタの読み出しは連続して行うことができます(マルチセクタ機能)。TC(ターミナルカウント)信号をFDCに入力することによってRead Dataコマンドは終了します。マルチセクタ機能はRead Deleted Data、Write Data、Write Deleted Dataコマンドにそれぞれ同様です。

　目的のセクタがインデックス信号を2回検出するまでに発見できないときは異常終了します。

```
 R/W                DATA
C     W   1  /MT /MF /SK / 0 / 0 / 1 / 1 / 0 /
          2  / - / - / - / - / - /HD /US1/US0/
          3                   C
          4                   H
          5                   R
          6                   N
          7                  EOT
          8                  GPL
          9                  DTL
E     R   -               転送データ
R     R   1                  ST0
          2                  ST1
          3                  ST2
          4                   C                  実行終了セクタのID情報
          5                   H
          6                   R
          7                   N
```

Read Deleted Data

Deleted Data とはなにか戸惑うかも知れませんが、単に読み込むセクタのDAM が DDAM になっただけで、それ以外は通常のセクタのリードと変わりません。普通使われることはありません。

```
 R/W                DATA
C      W   1  /MT /MF /SK / 0 / 1 / 1 / 0 / 0 /
          2-9         Read Dataと同じ
E      R   -             転送データ
R      R  1-7         Read Dataと同じ
```

Write Data

セクタのライトです。パラメータの説明は Read Data を参照して下さい。

```
 R/W                  DATA
C      W   1  /MT /MF / 0 / 0 / 0 / 1 / 0 / 1 /
           2  / - / - / - / - / - /HD /US1/US0/
           3                 C
           4                 H
           5                 R
           6                 N
           7                EOT
           8                GPL
           9                DTL
E      W   -             転送データ
R      R   1                ST0
           2                ST1
           3                ST2
           4                 C
           5                 H
           6                 R
           7                 N
```

Write Deleted Data

Deleted Data のライトです。パラメータは Read Data を参照して下さい。

```
 R/W                  DATA
C      W   1  /MT /MF / 0 / 0 / 1 / 0 / 0 / 1 /
          2-9        Write Dataと同じ
E      W   -             転送データ
R      R  1-7        Write Dataと同じ
```

Read ID

トラック上にあるセクタの ID だけを読み取ります。コマンドを実行してから一番始めに見つかったセクタの ID 情報をリザルトフェーズで返します。エラーでない ID がインデックスマークを 2 回検出するまで見つからない場合は、ST1 の ND ビットをセットして異常終了します。エラーでない ID が見つからずさらに ID アドレスマークが 1 回も検出されなかった時には、ND ビットの代わりに ST1 の MA バイトをセットして異常終了します。

```
 R/W                DATA
C     W   1  / 0 /MF / 0 / 0 / 1 / 0 / 1 / 0 /
          2  / - / - / - / - / - /HD /US1/US0/
E     -   -            データ転送はない
R     R   1                 ST0
          2                 ST1
          3                 ST2
          4                  C                   読み取った ID 情報
          5                  H
          6                  R
          7                  N
```

Write ID

トラックをフォーマットします。N でセクタ長を、SC で作成するセクタの数を指定します。セクタの内容は D で指定した値で埋められます。GPL はギャップ長ですが、通常は 54H を指定します。

エグゼキューションフェーズでは、セクタ毎に CHRN の 4 バイトのデータを FDC に転送します。つまり、4*SC バイトのデータを転送することになります。

```
 R/W                DATA
C     W   1  / 0 /MF / 0 / 0 / 1 / 1 / 0 / 1 /
          2  / - / - / - / - / - /HD /US1/US0/
          3                  N
          4                 SC
          5                 GPL
          6                  D
E     W   -         セクタ数分の ID 情報
R     R   1                 ST0
          2                 ST1
          3                 ST2
          4                  C
          5                  H
          6                  R
          7                  N
```

Read Diagnostic

　セクタリードなのですが、たとえCRCエラーが発生しても、セクタのIDと指定したID情報が異なっていても読み取り、セクタのIDで指定されるセクタ長は無視され、コマンドで指定したセクタ長分のデータが、EOTで指定したセクタ分読み込まれます。

```
 R/W                  DATA
C      W   1  / 0 /MF / 0 / 0 / 0 / 0 / 1 / 0 /
          2-9        Read Dataと同じ
E      R   -             転送データ
R      R  1-7        Read Dataと同じ
```

Scan Equal

　これと、Scan Low or Equal、Scan Low or Equalはディスク上のセクタの内容とCPUから送るデータを比較します。通常使用されません。

```
 R/W                  DATA
C      W   1  /MT /MF /SK / 1 / 0 / 0 / 0 / 1 /
          2-8        Read Dataと同じ
           9               STP
E      W   -            転送データ
R      R  1-7        Read Dataと同じ
```

Scan Low or Equal

```
 R/W                  DATA
C      W   1  /MT /MF /SK / 1 / 1 / 0 / 0 / 1 /
          2-9        Scan Equalと同じ
E      W   -            転送データ
R      R  1-7        Read Dataと同じ
```

Scan Low or Equal

```
 R/W                  DATA
C      W   1  /MT /MF /SK / 1 / 1 / 1 / 0 / 1 /
          2-9        Scan Equalと同じ
E      W   -            転送データ
R      R  1-7        Read Dataと同じ
```

Seek

シーク(ヘッドを目的のシリンダに移動すること)を行います。NCN でシリンダ番号を指定します。

Seek コマンドはコマンドフェーズでは FDC Busy 状態ですが、エグゼキューションフェーズでは Non-Busy 状態なので、他のドライブに対して
Seek、Recalibrate コマンドを実行させることができます。ただし、どれかのドライブが Busy 状態の時は FDC にリード／ライト系のコマンドを実行させてはいけません。

```
  R/W                   DATA
 C      W   1  / 0 / 0 / 0 / 0 / 1 / 1 / 1 / 1 /
            2  / - / - / - / - / - / - /US1/US0/
            3                    NCN
 E      -   -
```

Recalibrate

ヘッドをシリンダ 0(ドライブの Track0 信号が検出される)までシークします。

Recalibrate コマンドは Busy かどうかの状態について Seek と同様です。

```
  R/W                   DATA
 C      W   1  / 0 / 0 / 0 / 0 / 0 / 1 / 1 / 1 /
            2  / - / - / - / - / - / - /US1/US0/
 E      -   -
```

Sense Interrupt Status

Seek または Recalibrate コマンドによるシーク動作終了時のリザルトステータスを返します。PCN はシーク動作終了時のシリンダ番号を示します。

```
  R/W                   DATA
 C      W   1  / 0 / 0 / 0 / 0 / 1 / 0 / 0 / 0 /
 R      R   1                    ST0
            2                    PCN
```

Sense Device Status

FDC へ入力されるデバイス信号の状態を返します。

```
  R/W                   DATA
 C      W   1  / 0 / 0 / 0 / 0 / 0 / 1 / 0 / 0 /
            2  / - / - / - / - / - /HD /US1/US0/
 R      R   1                   ST3
```

Specify

FDC の各種時間用タイマ及び FDC の動作モードの初期値を FDC にセットします。

SRT(Step Rate Time)は、シーク動作時にドライブに送る STEP 信号の間隔時間を指定します。2ms 単位で 2〜32ms を指定できます。(0:32ms・・・15:2ms)

HUT(Head Unloaded Time)=1、HLT(Head Load Time)=1 を指定します。

ND(Non DMA Mode)は CPU と FDC の間のデータ転送方式を指定します。MSX では DMA 転送がサポートされないため、通常 1 を指定します。(0:DMA モード、1:Non DMA モード)

```
  R/W                   DATA
 C      W   1  / 0 / 0 / 0 / 0 / 0 / 0 / 1 / 1 /
            2  /      SRT      /      HUT      /
            3  /           HLT             /ND /
```

Invalid

定義されていないコマンドが与えられた時、または Sense Interrupt Status コマンドが実行された時シーク系コマンドによる INT 要求が発生していない場合です。

```
  R/W                   DATA
 C      W   1            その他のコード
 R      R   1                   ST0
```

7　FDC アクセスの例

765 系での FDC アクセスの例を示します。

```
STREG  ステータスレジスタ
DTREG  データレジスタ
```

FDC へのコマンド書き込み

HL レジスタの示すアドレスからにコマンドデータ列、B にコマンドのバイト数があるとします。

```
WRTCOM: LD      DE,07D0H        ;タイマ
                                ;(一定時間以内に BUSY=0 にならなければ異常終了)
WRTC01: LD      A,(STREG)       ;BUSY=0 になるのを待つ
        AND     10H
        JR      Z,WRTC02        ;NON BUSY になったらコマンド書き込みへ
        DEC     DE              ;タイマカウントダウン
        LD      A,D
        OR      E
        JR      NZ,WRTC01
        SCF                     ;異常終了
        RET

WRTC02: LD      A,(STREG)       ;DIO=0,RQM=1 になるのを待つ
        AND     0C0H
        CP      80H
        JR      NZ,WRTC02
        LD      A,(HL)          ;コマンドデータ
        LD      (DTREG),A       ;データレジスタに書き込む
        INC     HL
        DJNZ    WRTC02          ;コマンドのバイト数分繰り返す
        XOR     A               ;正常終了
        RET
```

リザルトステータスの受け取り

HL にリザルトステータスを格納するバッファのアドレスがあるとします。

```
GETRES: LD      A,(STREG)       ;ステータスレジスタを見る
        ADD     A,A
        JR      NC,GETRES       ;RQM=1 になるのを待つ
        RET     P               ;DIO=0 ならリザルトフェーズは終了
        LD      A,(DTREG)       ;データレジスタから読む
        LD      (HL),A          ;バッファに格納
        INC     HL
        JR      GETRES
```

CPU から FDC へのデータ転送(ライトデータなど)

HL レジスタに転送するデータの先頭アドレスがあるとします。

```
CPTOFD: LD      BC,STREG
        LD      DE,DTREG
CPTOF1: LD      A,(BC)
        ADD     A,A
        JP      NC,CPTOF1       ;RQM=1を待つ
        ADD     A,A
        RET     P               ;NDM=0になったら終了
        LD      A,(HL)          ;データ転送します
        LD      (DE),A
        INC     HL
        JP      CPTOF1
```

SEEK コマンド

C レジスタにヘッドを移動するシリンダの番号があるとします。

```
SEEK:   LD      HL,COMBUF       ;コマンドバッファ
        PUSH    HL
        POP     IX
        LD      (IX+00H),0FH    ;SEEKコマンド
        LD      (IX+01H),00H    ;ドライブ指定
        LD      (IX+02H),C      ;シリンダ番号
        LD      B,03H           ;コマンドのバイト数
        CALL    WRTCOM          ;コマンド書き込み

        LD      HL,COMBUF
        LD      (HL),08H        ;SENSE INTERRUPT STARUSコマンド
SEEK1:  LD      B,01H
        CALL    COMWRT          ;コマンド書き込み
        LD      HL,RESBUF       ;リザルトステータス用バッファ
        CALL    GETRES          ;リザルトステータス読み込み
        LD      A,(RESBUF)      ;ST0のSEビットが1になる(シーク終了)のを待つ
        BIT     5,A
        JR      Z,SEEK1
        AND     0C0H            ;正常終了か調べる
        RET     Z               ;正常終了
        SCF                     ;異常終了
        RET
```

WRITE DATA コマンド

```
WRTDAT: LD      HL,COMBUF
        PUSH    HL
        POP     IX
        LD      (IX+00H),45H    ;WRITE DATA コマンド
        LD      (IX+01H),00H    ;ドライブ、サイド
        LD      (IX+02H),00H    ;C
        LD      (IX+03H),00H    ;H
        LD      (IX+04H),01H    ;R
        LD      (IX+05H),02H    ;N
        LD      B,9
        CALL    WRTCOM          ;コマンド書き込み
        RET     C
        LD      HL,DATA         ;データ
        CALL    CPTOFD          ;データ転送
        CALL    TC              ;FDC にターミナルカウントを送るサブルーチン
        LD      HL,RESBUF
        CALL    GETRES          ;リザルトステータス受け取り
        LD      A,(RESBUF)      ;正常に終了したか調べる
        AND     0C0H
        RET     Z
        SCF
        RET
```

４．３　MB8876 系

１　MB8876

MB8876 系の FDC はウェスタンデジタル社が開発した FD1791 をもとに発展したもので、MB8876 は富士通の開発した製品です。もともと、片面、ドライブ 1 台用として設計されているため、レジスタ構成がシンプルである反面、2 台以上のドライブを接続すると余計な手間がかかります。

FD1791 の方が本家にあたるので FD1791 ファミリと呼ばれることもあります。

２　レジスタ

8876 系の FDC は外部からアクセスできるレジスタを 5 本持っています。

コマンドレジスタは書き込み専用の 8 ビットレジスタで、FDC にコマンドを実行させる時にコマンドを書き込みます。

ステータスレジスタは読み込み専用の 8 ビットレジスタで、FDC の状態等を表します。ステータスレジスタはコマンドレジスタと同じアドレスでアクセスします

データレジスタは、データ転送のために使用される 8 ビットレジスタです。

トラックレジスタはディスクのヘッドの位置を示す 8 ビットレジスタです。

セクタレジスタはセクタアクセス時にセクタ番号を指定する 8 ビットレジスタです。

ステータスレジスタは各ビットがそれぞれ意味を持っています。各ビットの意味は下に示します。ステータスレジスタの各ビットの意味はコマンドによって変化します。

ステータスレジスタの内容

	コマンド	STR7	STR6	STR5	STR4	STR3	STR2	STR1	STR0
TYPE I	すべてのコマンド	NOT READY	WRITE PROTECT	HEAD ENGAGED	SEEK ERROR	CRC ERROR	TRACK00	INDEX	BUSY
TYPE II	リードデータ	NOT READY	0	RECORD TYPE	RECORD N.FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY
	ライトデータ	NOT READY	WRITE PROTECT	WRITE FAULT	RECORD N.FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY
TYPE III	リードアドレス	NOT READY	0	0	RECORD N.FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY
	リードトラック	NOT READY	0	0	0	0	LOST DATA	DATA REQUEST	BUSY
	ライトトラック	NOT READY	WRITE PROTECT	WRITE FAULT	0	0	LOST DATA	DATA REQUEST	BUSY
TYPE IV	コマンド実行中	（今まで実行していたコマンドのステータスと同じ意味）							0
	コマンドなし	NOT READY	WRITE PROTECT	HEAD ENGAGED	0	0	TRACK00	INDEX	0
マスタリセット		（TYPE Iコマンドに準ずる）							

TYPE Iコマンド

STR0 BUSY	FDCがコマンド実行中であることを表します。
STR1 INDEX	インデックスホールを検出したことを示します。
STR2 TRACK00	ヘッドがトラック0の上にあることを示します。
STR3 CRC ERROR	IDフィールド読み出しの時にエラーがあったことを示します。
STR4 SEEK ERROR	ベリファイ動作が成功しなかったことを示します。
STR5 HEAD ENGAGED	ヘッドがメディアに押し付けられていることを示します。
STR6 WRITE PROTECT	ディスクへの書き込みが禁止されていることを示します。
STR7 NOT READY	ドライブが動作可能状態でないことを示します。

TYPE II/IIIコマンド

STR0 BUSY	FDCがコマンド実行中であることを表します。
STR1 DATA REQUEST	FDCがデータレジスタの読み出しまたは書き込みを要求していることを表します。
STR2 LOST DATA	データレジスタの読み出しまたは書き込みが所定の時間内に行われなかったことを表します。
STR3 CRC ERROR	IDフィールドまたはデータフィールドの読み出しの時にエラーがあったことを示します。
STR4 RECORD NOT FOUND	正しく読み出せ、指定されたIDをもつIDフィールドがなかったことを示します。
STR5 RECORD TYPE /WRITE FAULT	DAMがDDAMだった時にセットされます。 書き込み動作が打ち切られたことを示します。
STR6 WRITE PROTECT	ディスクへの書き込みが禁止されていることを示します。
STR7 NOT READY	ドライブが動作可能状態でないことを示します。

3　ドライブコントロール

FDC 自体にはドライブのモーターをコントロールするなどの機能はないので、これらは FDC とは別に外部回路があります。外部回路は機種によって異なる場合があります。

これら外部回路のコントロールのために FDC のレジスタの他にいくつかの I/O ポートを使用します。

その他に FDC から出ている信号のうち良く利用される物を説明します。

```
DRQ  DATA REQUEST  データレジスタへの書き込みまたは読み出し請求です。
IRQ  INTERRUPT     コマンドの終了、打ち切り及びタイプ IV コマンドの指定によって
      REQUEST      発生する出力信号です。
-MR  MASTER RESET  マスタリセットです。-MR=0 で全ての動作を停止します。
```

4　FDC のコントロール

FDC にある動作をさせるには、FDC のコマンドレジスタにコマンドを送ります。

コマンドは 11 種類あり、タイプ I からタイプ IV に大別されます。タイプ IV コマンドを除いてコマンドが実行中にはコマンドを書き込むことはできません。

FDC はコマンドレジスタにコマンドが書き込まれるとコマンドの実行を開始するので、コマンドの実行に必要なパラメータはコマンドを書き込む前に他のレジスタに書き込んでおく必要があります。

FDC のマスタリセット端子(MR)によって FDC にリセットを掛けることができます。この時内部のレジスタは次のようにプリセットされます。コマンドレジスタ=03H、ステータスレジスタ=(以前のステータスを保持)、データレジスタとトラックレジスタは不確定、セクタレジスタ=01H。マスタリセットが解除された時点で FDC はリストアコマンドを実行します。

5　FDC のコマンド

FDC のコマンドについて説明します。普通使用することのないと思われるコマンドについては説明を簡略化してあります。また、通常の MSX での使用を前提に記述してありますので、他の用途に使用する場合は参考文献などを参照して下さい。

TYPE I コマンド

リストア

トラック 0 へヘッドを移動します。

r1/r0 はステップレートフラグ(ヘッドを 1 トラックずつ移動する時間間隔)でドライブに適した値を指定します。値が大きいほど間隔が長くなります。h はヘッドロードフラグで実行開始時にヘッドをメディアにつける(h=1)か離すか(h=0)を指定しますが、ヘッドロード／アンロード機構がない場合が多いので、1 を指定して問題ありません。

V はトラック照合フラグです。V=1 ではヘッド移動後にディスク上の ID フィールドとトラックレジスタの値を比較します。ID フィールドが見つからなかったり、値が違ったりすると、ステータスレジスタの SEEK ERROR ビットをセットします。

コマンド　/ 0 / 0 / 0 / 0 / h / V / r1/ r0/

シーク

データレジスタで指定したシリンダにヘッドを移動します。

シークするシリンダの番号をデータレジスタにセットしておきます。

コマンド　/ 0 / 0 / 0 / 1 / h / V / r1/ r0/

ステップ

ヘッドを 1 トラック移動します。

u はトラックレジスタ更新フラグで、1 だとヘッドを移動すると同時にトラックレジスタも更新します。

コマンド　/ 0 / 0 / 1 / u / h / V / r1/ r0/

ステップイン

　ヘッドを1トラック内側に移動します。

コマンド　/ 0 / 1 / 0 / u / h / V / r1/ r0/

ステップアウト

　ヘッドを1トラック外側に移動します。

コマンド　/ 0 / 1 / 1 / u / h / V / r1/ r0/

TYPE II コマンド

リードデータ

　セクタを読み込みます。

　対象とするセクタのセクタ番号とトラック番号をセクタレジスタとトラックレジスタにセットしておかなければなりません。データ転送はデータレジスタを介して行われます。オプションでサイド番号をチェックすることができます。リード／ライトのバイト数はIDフィールドを読んだ時にIDのセクタ長指定バイトによって決定されます。

　Cは比較フラグで、1にするとサイド番号の比較を行います。その時はS(サイドフラグ)にサイド番号を指定します。サイド番号のLSBとSを比較します。

　他はm(マルチレコードフラグ)=0、E(ディレイフラグ)=0とすれば問題ありません。

コマンド　/ 1 / 0 / 0 / m / S / E / C / 0 /

ライトデータ

　セクタの書き込みです。セクタの指定などに関してはリードデータを見て下さい。

　a0はアドレスマークフラグで、1の時はDAMの代わりにDDAMを書き込みます。

コマンド　/ 1/ 0 / 1 / m / S / E / C / a0/

TYPE III コマンド

リードアドレス

　コマンドを指定してから最初に発見したIDの情報をデータレジスタを介してCPUに返します。返されるデータはIDの4バイトとIDCRC2バイトの計6バイトです。

```
コマンド　/1/1/0/0/0/E/0/0/
```

リードトラック

　インデックス信号を検出してから1トラックのデータを読み込みます。データの同期はインデックスマーク、IDアドレスマーク、データアドレスマークで行われます。

```
コマンド　/1/1/1/0/0/E/0/0/
```

ライトトラック

　1トラック分のデータをディスクに書き込みます。フォーマットのために、F5H、F6H、F7Hの3種類のデータはそのまま書き込まれずに特殊なデータとしてディスクに書き込まれます。

```
F5H　実際にはA1Hをミッシングクロックで書き込みます。
F6H　実際にはC2Hをミッシングクロックで書き込みます。
F7H　実際には内部で計算されたCRC2バイトを書き込みます。
```

```
コマンド　/1/1/1/1/0/E/0/0/
```

TYPE IV コマンド

フォースインタラプト

　このコマンドはいつでもコマンドレジスタに書き込み実行することができます。他のコマンドが実行中だとそのコマンドは打ち切られ、フォースインタラプトコマンドが実行されます。

　フォースインタラプトコマンドが実行されると、指定の条件でIRQが発生します。

I0～I3はIRQの発生条件を指定します。

```
I0=1  READY入力の立ち上りでIRQ発生
I1=1  READY入力の立ち下りでIRQ発生
I2=1  各インデックスパルスでIRQ発生
I3=1  無条件でただちにIRQ発生

コマンド  / 1 / 1 / 0 / 1 / I3/ I2/ I1/ I0/
```

6　フォーマットのデータ

　ディスクのフォーマットにはライトトラックコマンドを使います。ライトトラックコマンドではトラック上のデータをほぼそのまま書き込んでいくことになります。ここでは 2DD のディスクでの標準的なデータを示します。標準的な長さは 1 トラック 6250 バイトとなりますが、実際にはドライブの回転変動などで±2%までの誤差が出る可能性があります。
　セクタ部はセクタ数分(9 回)繰り返します。
　フォーマットでは、アドレスマークをミッシングクロックという特殊な形式で書き込まなければなりません。そのために F5H〜F7H のデータを使います。これらは FDC によって特殊な扱いを受けます。詳しくはライトトラックコマンドの説明を見てください。

フォーマットのデータ

```
  データ　バイト数　内容
    4EH      80
    00H      12
    F6H       3
    FCH       1     インデックスマーク
    4EH      50
セクタ部     (セクタ数分繰り返す)
    00H      12
    F5H       3
    FEH       1
シリンダ番号 1      (C:0〜4FH)
 サイド番号  1      (H:0〜1)
 セクタ番号  1      (R:1〜9)
  セクタ長   1      (N:2)
    F7H       1     ID CRC
    4EH      22
    00H      12
    F5H       3
    FBH       1     データマーク
    4EH     512     セクタの内容
    F7H       1     DATA CRC
    4EH      84
(ここまでセクタ部)
    4EH  トラックの終わりまで(ドライブの回転誤差で長さは変わります。)
```

7　FDC アクセスの例

8876 系での FDC アクセスの例を示します。

```
CMREG  コマンドレジスタ
STREG  ステータスレジスタ(=CMREG)
TRREG  トラックレジスタ
SCREG  セクタレジスタ
DTREG  データレジスタ
FDCST  FDC からの DRQ、IRQ 信号(b7:IRQ b6:-DRQ とする)
```

CPU から FDC へのデータ転送(ライトデータなど)

HL レジスタに、転送するデータのアドレスがあるとします。

```
CPTOFD: LD      BC,FDCST
        LD      DE,DTREG
CPTOF1: LD      A,(BC)          ;DRQ、IRQ の状態を読む
        ADD     A,A
        RET     C               ;IRQ がセットされたので終了
        JP      M,CPTOF1        ;DRQ がまだセットされていない
        LD      A,(HL)          ;メモリから読んで
        LD      (DE),A          ;データレジスタに書く
        INC     HL
        JP      CPTOF1
```

FDC から CPU へのデータ転送(リードデータなど)

HL レジスタに、データを転送するアドレスがあるとします。

```
FDTOCP: LD      BC,FDCST
        LD      DE,DTREG
FDTOC1: LD      A,(BC)          ;DRQ、IRQ の状態を読む
        ADD     A,A
        RET     C               ;IRQ がセットされたので終了
        JP      M,FDTOC1        ;DRQ がまだセットされていない
        LD      A,(DE)          ;データレジスタから読んで
        LD      (HL),A          ;メモリに書く
        INC     HL
        JP      FDTOC1
```

シーク

B にシークするシリンダ番号が入っているとします。

```
SEEK:   LD      A,B             ;シリンダ番号をデータレジスタにセット
        LD      (DTREG),A
        EX      (SP),HL         ;ウェイトを入れる
        EX      (SP),HL
        LD      A,11H           ;シークコマンド
        LD      (CMREG),A
        EX      (SP),HL
        EX      (SP),HL
SEEK1:  LD      A,(STREG)       ;BUSY=0 になるのを待つ
        RRA
        JR      C,SEEK1
```

ライトデータ

トラックレジスタにシリンダ番号、セクタレジスタにセクタ番号がセットされているとします。

```
RDDATA: LD      A,0A0H          ;ライトデータコマンドを書き込む
        LD      (CMREG),A
        LD      HL,DATA         ;データのアドレス
        CALL    CPTOFD
        LD      B,70H
RDDAT1: EX      (SP),HL
        EX      (SP),HL
        DJNZ    RDDAT1
        LD      A,(STREG)       ;ステータスレジスタを読む
        AND     0FCH
        RET                     ;成功なら Z=1
                                ;Z=0 はエラーが発生
                                ;       エラーの種類は A レジスタでわかる
```

４．４　トラックフォーマット

　トラック上に記録されているデータのフォーマットを説明します。
　2DD ディスクでは、ディスクの片面には 80 本のトラックがあり、それぞれのトラックに 8〜9 セクタのデータを記録しています。1 トラックの容量は 6250 バイトです。

6250(バイト)*80(トラック)*2(面)=1000000(バイト)で、2DD ディスクの容量が 1MB となっています。実際に記録される容量がこれよりも少ないのは、トラック上には実際のデータ以外にもいろいろ記録されているからです。ところで 6250 バイトというのは標準値です。トラックにデータを書き込む速度は一定なのですが、ディスクの回転は機種によって誤差があったり、同じマシンでも微妙に回転変動が生じているので、1 トラックに書き込みできる容量はその時々で変化します。ですから、トラック容量は最悪で 6250 バイトを中心に±2%程度の誤差が出る可能性を見込んでおかなければなりません。

　トラックは、プリアンブル部、セクタ部、ポストアンブル部の 3 つの領域で構成されます。

　プリアンブル部はトラックの先頭、つまりインデックス信号の検出される位置から存在します。この部分はトラックの先頭にある"余白"です。IAM(インデックスアドレスマーク)はトラックの先頭を示します。SYNC は IAM の読み取りのため同期を取るときに必要な部分です。実際には SYNC と IAM の先頭の 3 バイトのデータによって同期が取られます。IAM の始めの 3 バイトはミッシングクロックという特殊な形式でディスクに記録されています。GAP はデータとデータの間の余白です。回転変動等による影響を避けるためにも GAP が必要です。

　セクタ部は ID 部とデータ部に分かれています。　ID 部は個別のセクタを示すもので、セクタごとに存在します。IDAM(ID アドレスマーク)は、セクタの ID フィールドの始まりを示します。ID は C、H、R、N の 4 つのデータで構成されます。それぞれ、シリンダ番号、ヘッド番号、レコード番号、データ長を示します 2DD9 セクタフォーマットでは C は 0 から 79、H は 0〜1、R は 1〜9、N は 2 です。N はセクタの長さで、0:128 バイト、1:256 バイト、2:512 バイト、3:1024

バイト・・・、ですが、その上限はFDCによって異なる場合があります。N=3までは、どのFDCでも使用できます。4以上の場合FDCによって扱いが異なる場合があるので注意が必要です。

データ部は、実際のデータが記録される部分です。DAM(データアドレスマーク)がデータ部の始まりを示します。DAMの代わりにDDAM(デリーテッドアドレスマーク)も使用することができます。DAMとDDAMの違いは単にアドレスマークの値が違うというだけです。DAMが通常使用されます。DDAMで始まるセクタをどう処理するかはFDC、ディスクドライバによって異なります。
IDAMとDAM(DDAM)の始めの3バイトはIAMの場合と同様、ミッシングクロックで書き込まれています。SYNCとIDAM、DAM、DDAMで読み取りデータの同期を取ります。

CRCは記録されたデータを読んだ時に、正しくデータを読み込めたか(またはデータが正しく記録されたか)を調べるためのもので、チェックサムのようなものと考えると理解しやすいと思います。ID部とデータ部のそれぞれに存在し、それぞれIDCRC、DATACRCと呼ばれます。データの読み込み時にCRCエラーが発生すると、FDCはそれをシステムに知らせます。

GAP3の長さはデータ長によって異なることがあります。N=2では54Hバイトです。

ポストアンブル部はトラックの最後の余白です。この部分の長さはデータ部の長さによって変わりますし、またドライブの回転変動によっても変わります。

トラックの構成

		名称	データ	バイト数	
プリアンブル部		GAP0	4EH	80	
		SYNC	00H	12	
		IAM	C2H	3	（ミッシングクロック）
			FCH	1	
		GAP1	4EH	50	
セクタ部	ID部	SYNC	00H	12	セクタ部がセクタ数分繰り返す
		IDAM	A1H	3	（ミッシングクロック）
			FEH	1	
		C	C	1	
		H	H	1	
		R	R	1	
		N	N	1	
		CRC	CRC	2	
		GAP2	4EH	22	
	データ部	SYNC	00H	12	
		DAM/DDAM	A1H	3	（ミッシングクロック）
			FBH/F8H	1	
		データ	---	（ID部のNによって決まる）	
		CRC	CRC	2	
		GAP3	4EH	（データ長によって変化する）	
ポストアンブル部		GAP4	4EH	（変化する）	

第 6 部 APPENDIX

第１章　MSX ディスクシステム一覧

メーカー	機種名	ドライブ	カーネル	ドライバ	FDC	DISK ROM スロット
SONY	HBD-50	1DD*1				(拡張ドライブ)
	HBD-20W	2DD*1	3	S02	8876	(拡張ドライブ)
	HBD-30W	2DD*1		S01	8876	(拡張ドライブ)
	HBD-F1	2DD*1	3	S02	8876	(拡張ドライブ)
	HB-701FD	1DD*1				
	HB-F500	2DD*1				
	HB-F900	2DD*2	3	S01	8876	
	HB-T600	2DD*1				
	HB-F1XD	2DD*1	3	S02	8876	
	HB-F1XDmkII	2DD*1		S02	8876	
	HB-F1XDJ	2DD*1		S02	8876	
	HB-F1XV	2DD*1	3	S02	8876	
National	FS-FD351	2DD*1				(拡張ドライブ)
	CF-3300	2DD*1				
	FS-4600F	2DD*1	3	NA1	8876	
	FS-4700F	2DD*1	2	NA2	8876	
	FS-5500F1/F2	2DD*1	2	NA2	8876	3-3
	FS-5000F2	2DD*2	3	NA1	8876	3-3
Panasonic	FS-FD1	2DD*1	3	S04	8876	(拡張ドライブ)
	FS-FD1A	2DD*1	3	PA1	765	(拡張ドライブ)
	FS-A1F	2DD*1	3	PA2	765	3-2
	FS-A1FM	2DD*1				
	FS-A1WX	2DD*1	4	PA3	765	3-2
	FS-A1FX	2DD*1	4	PA3	765	3-2
	FS-A1WSX	2DD*1		PA3	765	3-2
	FS-A1ST(前期?)	2DD*1	6	PA4	765	3-2
	FS-A1ST(中期?)	2DD*1	6	PA6	765	3-2
	FS-A1ST(後期)	2DD*1	6	PA5	765	3-2
	FS-A1GT	2DD*1		PA5	765	3-2
SANYO	MFD-35	1DD*1				(拡張ドライブ)
	MPC-25F	1DD*1				
	MPC-25FD	2DD*1				
	MPC-25FK	2DD*1				
	PHC-77	2DD*1				
	PHC-70FD	2DD*1	4	SA1	765	3-2
	PHC-70FD2	2DD*2	4	SA1	765	3-2
CANON	V-30F	2DD*1	2	CA1		3-1
TOSHIBA	HX-F100	1DD*1				(拡張ドライブ)
	HX-F101	2DD*1				(拡張ドライブ)
	HX-34	2DD*1	2	T01	8876	
Victor	HC-F303	1DD*1				(拡張ドライブ)
	HC-90	2DD*1		VI1	8876	
	HC-95	2DD*2	3	VI1	8876	

MITSUBISHI	ML-30FD	2DD*1				(拡張ドライブ)
	ML-G30 Model1	2DD*1		MI1	8876	
	ML-G30 Model2	2DD*2		MI1	8876	
YAMAHA	FD-05	2DD*1				(拡張ドライブ)
	FD-03	1DD*1				(拡張ドライブ)
	YIS805/128	2DD*1				
	YIS805/256	2DD*2	1	YA1		0-2
HITACHI	MPF-310H	2DD*1				(拡張ドライブ)
	MB-H70	2DD*2				
DAISEN	MX30A	2DD				(拡張ドライブ)
	MX30B	2DD/2HD				(拡張ドライブ)
ASCII			7	AS1		HD I.F.

ドライブ数は基本的なシステムでの場合で、増設できる場合があります。

第２章　ファンクションコールクイックリファレンス

ファンクションコールのファンクション番号、名称、設定と戻り値のみを表にしてあります。名称と設定、戻り値の説明は、簡略化してある場合もあります。

この表は、ファンクションについて熟知している場合にのみ利用し、不明な点は第３部及び参考文献で確認して下さい。

	ファンクション名	設定/戻り値	
00H	プログラムの終了	なし	/なし
01H	コンソール入力	なし	/A 文字コード
02H	コンソール出力	E 文字コード	/なし
03H	補助入力	なし	/A 文字コード
04H	補助出力	E 文字コード	/なし
05H	プリンタ出力	E 文字コード	/なし
06H	直接コンソールI/O	E FFH:入力	/A 文字コード
		<>FFH:文字コード	/なし
07H	直接コンソール入力	なし	/A 文字コード
08H	エコーなしコンソール入力	なし	/A 文字コード
09H	文字列出力	DE 文字列のアドレス	/なし
0AH	バッファ行入力	DE バッファアドレス（DE） 最大文字数	/(DE+1) 入力文字数 (DE+2) 文字列
0BH	コンソールステータス	なし	/A 0FFH:入力あり 00H:なし
0CH	バージョンの獲得	なし	/HL 0022H
0DH	ディスクリセット	なし	/なし
0EH	ディスクの選択	E ドライブ番号(0〜7)	/なし
0FH	ファイルのオープン	DE FCBアドレス	/A 0:成功 FFH:失敗
10H	ファイルのクローズ	DE FCBアドレス	/A 0:成功 FFH:失敗
11H	最初のエントリの検索	DE FCBアドレス	/A 0:成功 FFH:失敗
12H	次のエントリの検索	なし	/A 0:成功 FFH:失敗
13H	ファイルの削除	DE FCBアドレス	/A 0:成功 FFH:失敗
14H	シーケンシャル読みだし	DE FCBアドレス	/A 0:成功 1:失敗
15H	シーケンシャル書き込み	DE FCBアドレス	/A 0:成功 1:失敗
16H	ファイルの作成	DE FCBアドレス	/A 0:成功 FFH:失敗
17H	ファイル名の変更	DE FCBアドレス (DE+17) 新ファイル名	/A 0:成功 FFH:失敗
18H	ログインベクタの獲得	なし	/HL ログインベクタ
19H	カレントドライブの獲得	なし	/A カレントドライブ(0〜7)
1AH	DTAのセット	DE DTAアドレス	/なし
1BH	アロケーション情報の獲得	省略	
21H	ランダム読みだし	DE FCBアドレス	/A 0:成功 1:失敗
22H	ランダム書き込み	DE FCBアドレス	/A 0:成功 1:失敗
23H	ファイルサイズの獲得	DE FCBアドレス	/A 0:成功 FFH:失敗
24H	ランダムレコードのセット	DE FCBアドレス	/なし
26H	ランダムブロック書き込み	DE FCBアドレス HL 書き込むレコード数	/A 0:成功 1:失敗
27H	ランダムブロック読みだし	DE FCBアドレス HL 読み込むレコード数	/A 0:成功 1:失敗 HL 読んだレコード数

```
        ファンクション名                設定/戻り値

28H  ランダム書き込みゼロフィル  DE  FCBアドレス    /A  0:成功  1:失敗
2AH  日付の獲得               なし               /HL:年  D:月  E:日  A:曜日
2BH  日付のセット              HL:年  D:月  E:日    /A  0:成功  FFH:無効
2CH  時刻の獲得               なし
                         /H:時  L:分  D:秒  E:1/100秒
2DH  時刻のセット              H:時  L:分  D:秒  E:1/100秒
                         /A  0:成功  FFH:無効
2EH  ベリファイフラグセット      E  0:無効  0以外:有効/なし
2FH  セクタ読みだし  DE  セクタ番号  L  ドライブ  H  セクタ数  /A  0:成功
30H  セクタ書き込み  DE  セクタ番号  L  ドライブ  H  セクタ数  /A  0:成功
31H  ディスクパラメータの獲得  DE  バッファのアドレス  /A  エラーコード
                          L  ドライブ番号          /DE  保存される
40H  最初のエントリの検索     DE  ASCIIZ,FIB  HL  ASCIIZ文字列(DE=FIB)
                        B  検索属性      IX  新しいFIB
                       /A  エラーコード (IX)  一致するエントリ
41H  次のエントリの検索       IX  FIB  /A  エラー  (IX)  次の一致するエントリ
42H  新しいエントリの検索     DE  ASCIIZ,FIB  HL  ASCIIZ文字列(DE=FIB)
                        B  検索属性,新規作成フラグ(b7)
                        IX  テンプレートファイル名を保持している新しいFIB
                       /A  エラー  (IX)  新しいエントリ
43H  ファイルハンドルのオープン  DE  ASCIIZ,FIB  A  オープンモード
                          /A  エラー        B  新しいファイルハンドル
44H  ファイルハンドルの作成      DE  ASCIIZ      A  オープンモード
                           B  要求する属性/新規作成フラグ(b7)
                          /A  エラー        B  新しいファイルハンドル
45H  ファイルハンドルのクローズ  B  ファイルハンドル  /A  エラー
46H  ファイルハンドルの確保      B  ファイルハンドル  /A  エラー
47H  ファイルハンドルの複製      B  ファイルハンドル  /A  エラー
                                                B  新しいファイルハンドル
48H  ファイルハンドルからの読み出し  B  ファイルハンドル  DE  バッファアドレス
                              HL  読み込むバイト数
                             /A  エラー  HL  実際に読み込んだバイト数
49H  ファイルハンドルへの書き込み    B  ファイルハンドル  DE  バッファアドレス
                              HL  書き込むバイト数
                             /A  エラー  HL  実際に書き込んだバイト数
4AH  ファイルハンドルポインタの移動  B  ファイルハンドル  A  方式コード
                              DE:HL  符号付きオフセット
                             /A  エラー  DE:HL  新しいファイルポインタ
4BH  デバイスのI/O制御        B  ファイルハンドル  A  サブファンクションコード
                        DE  他のパラメータ
                       /A  エラー  DE  他の結果
4CH  ファイルハンドルのテスト    B  ファイルハンドル  DE  ASCIIZ,FIB
                         /A  エラー  B  0:同じファイルでない  FFH:同じ
4DH  ファイル・サブディレクトリの削除      DE  ASCIIZ,FIB /A  エラー
4EH  ファイル名・サブディレクトリ名の変更  DE  ASCIIZ,FIB  HL  ASCIIZ /A エラー
4FH  ファイル・サブディレクトリの移動      DE  ASCIIZ,FIB  HL  ASCIIZ /A エラー
50H  ファイル属性の獲得・セット          DE  ASCIIZ,FIB  A  0:獲得 1:セット
                                   L  新しい属性バイト(A=1の場合)
                                  /A  エラー  L  現在の属性バイト
```

ファンクション名　　　　　　　　　設定/戻り値

51H　ファイルの日付および時刻の獲得・セット　DE　ASCIIZ,FIB　A　0:獲得 1:セット
IX　新しい時刻(A=1)　HL　新しい日付(A=1)
/A　エラー　DE　時刻　HL　日付
52H　ファイルハンドルの削除　　　　B　ファイルハンドル　/A　エラー
53H　ファイルハンドルの名前の変更　　B　ファイルハンドル　HL　ASCIIZ /A　エラー
54H　ファイルハンドルの移動　　　　B　ファイルハンドル　HL　ASCIIZ /A　エラー
55H　ファイルハンドルの属性の獲得・セット B ファイルハンドル　A 0:獲得 1:セット
L　新しい属性バイト(A=1)
/A　エラー　　L　現在の属性バイト
56H　ファイルハンドルの日付及び時刻の獲得・セット　B　ファイルハンドル
A　0:獲得 1:セット　IX　時刻(A=1)　HL　日付(A=1)
/A　エラーDE　時刻　HL　日付
57H　ディスク転送アドレスの獲得　　なし　/DE　現在のディスク転送アドレス
58H　ベリファイフラグ設定の獲得　　なし　/B　00H:無効 FFH:有効
59H　カレントディレクトリの獲得　　B　ドライブ番号　DE　64バイトバッファ
/A　エラー　DE　カレントパス
5AH　カレントディレクトリの変更　　DE　ASCIIZ　/A　エラー
5BH　パス名の解析　B　ボリューム名フラグ(b4)　DE　ASCIIZ　/A　エラー
DE　終了文字へのポインタ　HL　最後の項目の先頭
B　解析フラグ　C　論理ドライブ
5CH　ファイル名の解析　DE　ASCIIZ　HL　11バイトバッファ
/A　エラー　DE　終了文字　HL　保存　B　解析フラグ
5DH　文字の検査　D　文字フラグ　E　検査する文字/　A　0　D　変更された文字フラグ
E　検査された(大文字にされた)文字
5EH　完全なパス文字列の獲得　DE　64バイトバッファ
/A　エラー　(DE)　完全なパス文字列
HL　最後の項目の初めへのポインタ
5FH　ディスクバッファのフラッシュ　B　ドライブ番号(0FFH:全て)
D　00H:フラッシュのみ　FFH:フラッシュして無効にする
/A　エラー
60H　子プロセスの起動　　なし　　　　　　/A　エラー　B　親プロセスのID
61H　親プロセスに戻る　　B　親のプロセスID /A　エラー
B　子プロセスからの1次エラーコード
C　子プロセスからの2次エラーコード
62H　エラーコードを伴った終了　　　　B　終了のエラーコード　/なし
63H　アボート終了ルーチンの定義　　　DE　アボート終了ルーチン(0:解除)　/A 0
64H　ディスクエラー処理ルーチンの定義　DE　ディスクエラールーチン(0:解除)/A 0
65H　直前のエラーコードの獲得　　　　なし　/A 0　B　直前のエラーコード
66H　エラーコードの説明　　B　説明すべきエラーコード　DE　64バイトのバッファ
/A 0　B 0あるいは変更無し
(DE)　エラーメッセージが入る
67H　ディスクのフォーマット　B　ドライブ番号　HL　バッファ　DE　バッファサイズ
A　0:選択文字列 1〜9 フォーマット
FEH,FFH　ブートセクタの更新
/A　エラー　B　スロット(エントリでA=0)
HL　アドレス(エントリでA=0)
68H　RAMディスクの作成あるいは消去　B　0:消去 1〜FEH:作成 FFH:サイズを返す
/A　エラー　B　RAMディスクのサイズ
69H　セクタバッファの割り付け　B　0:バッファ数を返す 1〜FFH:要求するバッファ数
/A　エラー　B　バッファの現在の数

```
       ファンクション名               設定/戻り値

6AH  論理ドライブの割り当て           B  論理ドライブ番号(1:A・・・)
                                      D  物理ドライブ番号(1:A・・・)
                                     /A  エラー  D  物理ドライブ番号(1:A・・・)
6BH  環境変数の獲得  HL  ASCIIZ  DE  バッファ  B  バッファサイズ
                   /A  エラー  DE  保存される、A=0 の場合バッファが満たされる
6CH  環境変数のセット  HL  ASCIIZ  DE  ASCIIZ  /A  エラー
6DH  環境変数の検索  DE  環境変数番号  HL  名前文字列のバッファへのポインタ
                    B  バッファサイズ
                   /A  エラー  HL  保存され、バッファが満たされる
6EH  ディスク検査ステータスの獲得・セット     A  00H:獲得 01H:セット
                                            B  00H:有効(A=01H) FFH:無効(A=01H)
                                           /A  エラー  B  現在の設定
6FH  MSX-DOS のバージョン番号の獲得      なし  /A  0  BC  カーネルバージョン
                                            DE  MSXDOS2.SYS のバージョン
70H  リダイレクションの状態の獲得・セット A  00H:獲得 01H:セット
                         B  新しい状態 b0:標準入力  b1:標準出力
                        /A  エラー  B  コマンド以前のリダイレクションの状態
```

編集後記

テクニカルガイドブックディスク編が第二版となりました。この第二版は実際のところ初版からあまり変わっていないので、初版を持っている方にはたいして足しにならないかも知れません。初版で不十分だった部分をいろいろ入れたいとは思っていたのですが、結果的にはたいしたことはできませんでした。

テクニカルガイドブックディスク編という名称は長くて面倒ですが、公式の通称(？)はディスクテクガイです。しかし中にはテクガイディスク編と呼んでいるという方もいるようです。好きなように呼んで下さい。

ディスクテクガイは安心して利用できる資料を目標にしています。間違いを発見された場合や、こうした方がいいと思った場合などぜひご連絡下さい。

ディスクテクガイに関してのご感想やご意見、情報などは筆者宛でお送り下さい。質問などの場合は必ず返信封筒を同封の上でお願いします。

テクニカルガイドブックディスク編の独特かつ魅力的な扉イラストは、その画風で好評の長良川鵜さんの手によるものです。今回、第二版のために新たに描いて頂きました。初版と同様、本書の重要な引き立て役となっています。

著者略歴

和泉澤　拓。1973年6月千葉県生まれ。CF-3000に始まり、FS-5500、FS-A1STとひたすらMSXをさわっていた。現在はDynabook、X68Kユーザーでもある。バックアップ活用テクニック誌パート20の大コンテストで"COPY CAT"が準グランプリ受賞。現在はX68K上でプログラムを書いている。千葉大学理学部在学中。

イラスト・長良川　鵜

MSXテクニカルガイドブック　ディスク編 検印省略
1992年12月30日　初版発行
1993年6月15日　第二版発行
1996年11月24日　HTML版発行
発行所　アスキャット

著者　和泉澤　拓

HTML化　JR SOFT

(注)編集後記は製品版のものです。